FUJITSU MICROELECTRONICS
CONTROLLER MANUAL

CM71-10146-1E

FR80

32-BIT MICROCONTROLLER

MB91660 Series

HARDWARE MANUAL

FUJITSU

FR80

32-BIT MICROCONTROLLER

MB91660 Series

HARDWARE MANUAL

The information for microcontroller supports is shown in the following homepage.
Be sure to refer to the "Check Sheet" for the latest cautions on development.

"Check Sheet" is seen at the following support page

"Check Sheet" lists the minimal requirement items to be checked to prevent problems beforehand in system development.

http://edevice.fujitsu.com/micom/en-support/

FUJITSU MICROELECTRONICS LIMITED

Preface

Thank you for your continued use of Fujitsu microelectronics semiconductor products.
Read this manual and "Data Sheet" thoroughly before using products in the MB91660 series.

■ Purpose of this manual and intended readers

This manual explains the functions and operations of the MB91660 series and describes how it is used. The manual is intended for engineers engaged in the actual development of products using the MB91660 series.

Note: FR is an abbreviation for the FUJITSU RISC controller, which is a product of Fujitsu Microelectronics Limited.

■ Trademark

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ License

Purchase of Fujitsu I^2C components conveys a license under the Philips I^2C Patent Rights to use, these components in an I^2C system provided that the system conforms to the I^2C Standard Specification as defined by Philips.

■ Sample programs and development environment

Fujitsu Microelectronics offers sample programs free of charge for using the peripheral functions of the FR80 family. Fujitsu Microelectronics also makes available descriptions of the development environment required for the MB91660 series. Feel free to use them to verify the operational specifications and usage of this Fujitsu microelectronics microcontroller.

- Microcontroller support information:

  http://edevice.fujitsu.com/micom/en-support/

* Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.

Fujitsu microelectronics assumes no responsibility for any damage that may occur as a result of using a sample program.

- The contents of this document are subject to change without notice.
  Customers are advised to consult with sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of FUJITSU MICROELECTRONICS device; FUJITSU MICROELECTRONICS does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. FUJITSU MICROELECTRONICS assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of FUJITSU MICROELECTRONICS or any third party or does FUJITSU MICROELECTRONICS warrant non-infringement of any third-party's intellectual property right or other right by using such information. FUJITSU MICROELECTRONICS assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
  Please note that FUJITSU MICROELECTRONICS will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- Exportation/release of any products described in this document may require necessary procedures in accordance with the regulations of the Foreign Exchange and Foreign Trade Control Law of Japan and/or US export control laws.
- The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

Copyright ©2008 FUJITSU MICROELECTRONICS LIMITED All rights reserved.

Manuals That Make Up the Manuals for This Series

The manuals used for this series are listed below. See the manual appropriate to the applicable conditions.
The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

■ Hardware manual

- FR80 FAMILY MB91660 SERIES HARDWARE MANUAL (CM71-10146-1E) (this manual)

■ Data sheet

- MICROCONTROLLER 32-bit ORIGINAL FR80 FAMILY MB91660 SERIES DATA SHEET (DS07-16905-1E)

■ Programming manual

- FR80 FAMILY PROGRAMMING MANUAL (CM71-00102-1E)

  This manual explains a programming model and instructions for the FR80 family CPUs.

■ Hardware tool-related manual

- DSU-FR EMULATOR MB2198-01 HARDWARE MANUAL (CM71-00413-2E)

  This manual explains emulator handling and specifications, and it explains how to connect and operate the emulator.

■ Software tool-related manuals

- SOFTUNE™ WORKBENCH OPERATION MANUAL for V6 (CM71-00328-3E)

  This manual explains how to operate the integrated development environment called SOFTUNE and the development procedures.

- SOFTUNE™ WORKBENCH USER'S MANUAL for V6 (CM71-00329-3E)

  This manual explains the basic functions and dependent functions of SOFTUNE Workbench.

- SOFTUNE™ WORKBENCH COMMAND REFERENCE MANUAL for V6 (CM71-00330-3E)

  This manual explains the commands and built-in variables/functions of SOFTUNE Workbench.

- FR FAMILY 32-BIT MICROCONTROLLER EMBEDDED C PROGRAMMING MANUAL FOR APPLICATION (CM71-00324-1E)

  This manual describes the know-how for creating built-in systems using the C compiler fcc911 for the FR family. The manual explains how to create efficient C programs using the architecture of the FR family and provides the notes.

- FR FAMILY SOFTUNE C/C++ COMPILER MANUAL for V6 (CM81-00206-3E)

  Refer to this manual when using SOFTUNE C/C++ compiler to create/develop application programs in C and C++.

- FR FAMILY SOFTUNE™ ASSEMBLER MANUAL for V6 (CM71-00203-3E)

  This manual explains the functions of Fujitsu SOFTUNE™ Assembler operating in Windows 98, Windows Me, Windows 2000, or Windows XP and how to use it.

- SOFTUNE™ LINKAGE KIT MANUAL for V6 (CM71-00327-3E)

  This manual explains the functions of Fujitsu SOFTUNE™ Linkage Kit operating in Windows 98, Windows Me, Windows 2000, or Windows XP and how to use it.

  See the manual when developing an application program.
- FR Family ABSOLUTE ASSEMBLY LIST GENERATOR TOOL MANUAL (CM71-00305-2E)

  This manual explains absolute assemble lists.
- FR-V/FR FAMILY SOFTUNE C/C++ ANALYZER MANUAL for V5 (CM81-00309-4E)

  This manual explains the functions of C/C++ Analyzer and how to use it.
- FR-V/FR FAMILY SOFTUNE C/C++ CHECKER MANUAL for V5 (CM81-00310-4E)

  This manual explains the functions of C/C++ Checker and how to use it.

■ REALOS-related manuals

● REALOS µITRON3.0-related manuals

- FR/F^{2}MC FAMILY IN CONFORMANCE WITH µITRON SPECIFICATIONS SOFTUNE™ REALOS™/FR/907/896 CONFIGURATOR MANUAL (CM71-00322-5E)

  This manual explains the functions and operations of SOFTUNE REALOS Configurator.
- FR-V/FR/F^{2}MC FAMILY IN CONFORMANCE WITH µITRON SPECIFICATIONS SOFTUNE™ REALOS™/ANALYZER MANUAL (CM81-00315-3E)

  This manual explains the functions provided by SOFTUNE REALOS Analyzer and how to utilize the functions.
- FR FAMILY IN CONFORMANCE WITH µITRON 3.0 SPECIFICATIONS SOFTUNE REALOS/ FR USER'S GUIDE (CM71-00320-3E)

  This manual explains the configuration/activation of REALOS/FR application systems.

  See the manual when performing comprehensive work for an entire system.
- FR FAMILY IN CONFORMANCE WITH µITRON 3.0 SPECIFICATIONS SOFTUNE REALOS/ FR KERNEL MANUAL (CM71-00321-3E)

  This manual explains the functions provided by SOFTUNE REALOS/FR and how to utilize the functions.

  See the manual when creating an application system or user program.

● REALOS µITRON4.0-related manuals

- FR FAMILY IN CONFORMANCE WITH µITRON 4.0 SPECIFICATIONS SOFTUNE™ REALOS™/FR Spec.4 PROGRAMMING MANUAL (CM81-00316-1E)

  This manual explains the functions provided by SOFTUNE REALOS/FR Spec.4 and how to utilize the functions.
- FR-V/FR FAMILY IN CONFORMANCE WITH µITRON 4.0 SPECIFICATIONS SOFTUNE™ REALOS™ KERNEL MANUAL (CM81-00312-2E)

  This manual explains the functions provided by SOFTUNE REALOS/FRV/FR Spec.4 and how to utilize the functions.
- FR-V/FR FAMILY IN CONFORMANCE WITH µITRON 4.0 SPECIFICATIONS SOFTUNE™ REALOS™ CONFIGURATOR MANUAL (CM81-00311-3E)

  This manual explains the functions provided by SOFTUNE REALOS Configurator (GUI) and how to utilize the functions.

- FR-V/FR /F^2MC FAMILY IN CONFORMANCE WITH µITRON SPECIFICATIONS SOFTUNE REALOS^{TM} ANALYZER MANUAL (CM81-00315-3E)

  This manual explains the functions provided by SOFTUNE REALOS Analyzer and how to utilize the functions.

How to Use This Manual

■ Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- **Search from the table of the contents**

  The table of the contents lists the manual contents in the order of description.

- **Search from the register list**

  The register list lists all the registers of this device. You can look up the name of a desired register on the list to find the address of its location or the page that explains it.

  The address where each register is located is not described in the text. To verify the address of a register, see "APPENDIX A I/O Map", and "APPENDIX B List of Registers".

- **Search from the index**

  You can look up the keyword such as the name of a peripheral function in the index to find the explanation of the function.

■ About the chapters

Basically, this manual explains 1 peripheral function per chapter.

■ Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

CONTENTS

CHAPTER 1 Overview 1
1.1 MB91660 Series Overview 2
1.2 MB91660 Series Product Configuration 7
1.3 MB91660 Series Block Diagram 9
1.4 Package Dimensions 10

CHAPTER 2 Pins of the MB91660 Series 11
2.1 Pin Assignment Diagram 12
2.2 Pin Functions 13
2.3 I/O Circuit Types 38
2.4 Setting Method for Pins 44

CHAPTER 3 CPU 91
3.1 Memory Space 92
3.2 Features of the Internal Architecture 94
3.3 Operation Modes 95
3.4 Pipeline 96
3.5 Overview of Instructions 98
3.5.1 Arithmetic Operation 98
3.5.2 Load and Store 98
3.5.3 Branch 99
3.5.4 Logical Operation and Bit Operation 99
3.5.5 Direct Addressing 99
3.5.6 Bit Search 99
3.5.7 Other 99
3.6 Basic Programming Model 100
3.7 Registers 101
3.7.1 General-purpose Registers (R0 to R15) 101
3.7.2 Program Status Register (PS) 102
3.7.3 Program Counter (PC) 107
3.7.4 Table Base Register (TBR) 108
3.7.5 Return Pointer (RP) 109
3.7.6 System Stack Pointer (SSP) 110
3.7.7 User Stack Pointer (USP) 111
3.7.8 Multiply & Divide Registers 112
3.8 Data Configuration 113
3.8.1 Bit Ordering 113
3.8.2 Byte Ordering 114
3.8.3 Word Alignment 115
3.9 Addressing 116
3.9.1 Direct Addressing Areas 116
3.9.2 20-bit Addressing Area 117
3.9.3 32-bit Addressing Area 117
3.9.4 Vector Table Initial Area 117

3.10 Branch Instructions 118
3.10.1 Operation with Delay Slots 118
3.10.2 Operation without Delay Slots 120
3.11 EIT (Exception, Interrupt, Trap) 121
3.11.1 EIT Sources 121
3.11.2 Return from EIT 121
3.11.3 Interrupt Level 122
3.11.4 I Flag 123
3.11.5 Interrupt Level Mask Register (ILM) 124
3.11.6 Level Mask for Interrupts 124
3.11.7 Interrupt Control Register (ICR) 125
3.11.8 System Stack Pointer (SSP) 125
3.11.9 Interrupt Stack 125
3.11.10 Table Base Register (TBR) 126
3.11.11 EIT Vector Table 126
3.11.12 Multi-EIT Processing 127
3.11.13 Operation 128
3.11.14 INT Instruction Operation 129
3.11.15 INTE Instruction Operation 130
3.11.16 Step Trace Trap Operation 130
3.11.17 Undefined Instruction Exception Operation 131
3.11.18 RETI Instruction Operation 131
3.11.19 Delay Slots and EIT 131

CHAPTER 4 Clock Generating Parts 133
4.1 Overview 134
4.2 Configuration 135
4.2.1 Clock Generating Parts 135
4.2.2 Source Clock (SRCCLK) Selection Block 138
4.3 Pins 139
4.4 Registers 140
4.4.1 Clock Source Select Register (CSELR) 141
4.4.2 Clock Source Monitor Register (CMONR) 145
4.4.3 Clock Stabilization Time Select Register (CSTBR) 148
4.4.4 PLL Configuration Register (PLLCR) 151
4.5 Explanation of Operations 155
4.5.1 Explanation of Clock Source Operations 155
4.5.2 Switching the Source Clock (SRCCLK) 158
4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK) 161

CHAPTER 5 Clock Division Control Part 163
5.1 Overview 164
5.2 Internal Clocks 165
5.3 Configuration 167
5.4 Registers 168
5.4.1 Divide Clock Configuration Register 0 (DIVR0) 169
5.4.2 Divide Clock Configuration Register 1 (DIVR1) 171
5.4.3 Divide Clock Configuration Register 2 (DIVR2) 173
5.5 Division Rate 175

CHAPTER 6 Main Timer 177
6.1 Overview 178
6.2 Configuration 179
6.3 Registers 180
6.3.1 Main Timer Control Register (MTMCR) 181
6.4 Interrupts 185
6.5 An Explanation of Operations and Setting Procedure Examples 186
6.5.1 Main Timer Operation 186
6.5.2 Transition to Stop Mode 188

CHAPTER 7 Sub Timer 189
7.1 Overview 190
7.2 Configuration 191
7.3 Registers 192
7.3.1 Sub Timer Control Register (STMCR) 193
7.4 Interrupts 197
7.5 An Explanation of Operations and Setting Procedure Examples 198
7.5.1 Sub timer operation 198
7.5.2 Transition to Stop Mode, and Watch Mode 200

CHAPTER 8 Low-power Dissipation Mode 201
8.1 Overview 202
8.2 Configuration 203
8.3 Registers 205
8.3.1 Standby Mode Control Register (STBCR) 206
8.3.2 Sleep Rate Configuration Register (SLPRR) 209
8.4 An Explanation of Operations and Setting Procedure Examples 211
8.4.1 Operation When Clock Control Is Set 212
8.4.2 Operation in Doze Mode 214
8.4.3 Operation in Sleep Mode 215
8.4.4 Operation in Main Timer Mode 218
8.4.5 Operation in Watch Mode 220
8.4.6 Operation in Stop Mode 222
8.5 Notes on Use 225

CHAPTER 9 Reset 227
9.1 Overview 228
9.2 Configuration 229
9.3 Pins 231
9.4 Registers 232
9.4.1 Reset Result Register (RSTRR) 233
9.4.2 Reset Control Register (RSTCR) 235
9.5 Explanation of Operations 237
9.5.1 Reset Types 237
9.5.2 Reset Resource 238
9.5.3 Operation of Reset 239
9.6 Operating State and Transition 244

CHAPTER 10 Interrupt Controller 249
10.1 Overview 250
10.2 Configuration 251
10.3 Registers 252
10.3.1 Interrupt Control Register (ICR00 to ICR47) 253
10.4 An Explanation of Operations and Setting Procedure Examples 255
10.4.1 Explanation of Operations of Interrupt Controller 255
10.5 Notes on Use 257

CHAPTER 11 Interrupt Request Batch-Read Function 259
11.1 Overview 260
11.2 Configuration 261
11.3 Registers 262
11.3.1 Interrupt Request Batch-Read Register 0 Upper (IRPR0H) 263
11.3.2 Interrupt Request Batch-Read Register 1 Upper/Lower (IRPR1H/ IRPR1L) 265
11.3.3 Interrupt Request Batch-Read Register 2 Upper (IRPR2H) 268
11.3.4 Interrupt Request Batch-Read Register 2 Lower (IRPR2L) 269
11.3.5 Interrupt Request Batch-Read Register 3 Upper (IRPR3H) 270
11.3.6 Interrupt Request Batch-Read Register 3 Lower (IRPR3L) 271
11.3.7 Interrupt Request Batch-Read Register 4 Upper (IRPR4H) 272
11.3.8 Interrupt Request Batch-Read Register 4 Lower (IRPR4L) 273
11.3.9 Interrupt Request Batch-Read Register 5 Upper (IRPR5H) 275
11.3.10 Interrupt Request Batch-Read Register 5 Lower (IRPR5L) 277
11.3.11 Interrupt Request Batch-Read Register 6 Upper (IRPR6H) 279
11.3.12 Interrupt Request Batch-Read Register 6 Lower (IRPR6L) 281
11.3.13 Interrupt Request Batch-Read Register 7 Upper (IRPR7H) 283
11.3.14 Interrupt Request Batch-Read Register 7 Lower (IRPR7L) 285
11.4 Notes on Use 286

CHAPTER 12 Delay Interrupt 287
12.1 Overview 288
12.2 Configuration 289
12.3 Registers 290
12.3.1 Delayed Interrupt Control Register (DICR) 291
12.4 An Explanation of Operations and Setting Procedure Examples 292
12.4.1 Explanation of Delay Interrupt Operations 292
12.5 Notes on Use 293

CHAPTER 13 External Bus Interface 295
13.1 Overview 296
13.2 Configuration 298
13.3 Pins 300
13.4 Registers 302
13.4.1 Area Setting Registers (ASR0 to ASR3) 303
13.4.2 Area Configuration Registers (ACR0 to ACR3) 306
13.4.3 Area Wait Registers (AWR0 to AWR3) 309
13.4.4 DMA Transfer Registers (DMAR0 to DMAR3) 318
13.5 Protocols 320
13.5.1 Address Data Split Bus Protocol 320

13.5.2 Address Data Multiplex Bus Protocol 326
13.6 Timing Settings 332
13.6.1 Read Access Automatic Wait 333
13.6.2 Write Access Automatic Wait 336
13.6.3 Read Access Idle Cycle 339
13.6.4 Write Recovery Cycle 342
13.6.5 Read Access Setup Cycle 345
13.6.6 Read Access Hold Cycle 347
13.6.7 Write Access Setup Cycle 350
13.6.8 Write Access Hold Cycle 352
13.6.9 Chip Select Delay Cycle 354
13.6.10 Address Output Extension Cycle 357
13.6.11 Address Strobe Output Extension Cycle 359
13.7 Access Cycle Extension Using the RDY Pin 362
13.8 Number of Access Cycles 365
13.9 Address Information and Address Alignment 367
13.9.1 Address Information 367
13.9.2 Address Alignment 368
13.10 Data Alignment 369
13.10.1 Big Endian 371
13.10.2 Little Endian 375
13.11 External Bus DMA Transfer 379
13.11.1 Transfer Requests with the DREQ0 to DREQ3 Pins 380
13.11.2 Transfer Request Acceptance with the DACK0 to DACK3 Pins 384
13.11.3 Transfer End Signal by the DEOP0 to DEOP3 Pins 386
13.12 CS Area Setting Procedure 387

CHAPTER 14 Slave Interface 393
14.1 Overview 394
14.2 Configuration 395
14.3 Pins 397
14.4 Registers 399
14.4.1 Control Register (SLIFCR) 400
14.4.2 Status Register (SLIFSR) 404
14.4.3 Reception FIFO Status Register (SLIFRXSR) 408
14.4.4 Transmission FIFO Status Register (SLIFTXSR) 410
14.4.5 Reception FIFO Data Register (SLIFRXDR) 411
14.4.6 Transmission FIFO Data Register (SLIFTXDR) 412
14.4.7 Reception Register (SLIFRXR) 413
14.4.8 Transmission Register (SLIFTXR) 414
14.4.9 DMA Reception FIFO Register (SLIFDRXR) 415
14.5 Interrupts 416
14.6 Explanation of Operations and Setting Procedure Examples 418
14.6.1 Explanation of Operations of the Slave Interface 418
14.7 Notes on Use 423

CHAPTER 15 I/O Ports 425
15.1 Overview 426
15.2 Configuration 428

15.3 Pins 433
15.4 Registers 434
15.4.1 Port Data Direction Registers (DDR0 to DDRK) 436
15.4.2 Port Function Registers (PFR0 to PFRH) 439
15.4.3 Extended Port Function Registers (EPFR0 to EPFR35) 442
15.4.4 Port Data Registers (PDR0 to PDRK) 463
15.4.5 Pull-up Resistor Control Registers (PCR0 to PCRK) 465
15.4.6 A/D Channel Enable Register (ADCHE) 467
15.5 Notes on Use 468

CHAPTER 16 External Interrupt Controllers 471
16.1 Overview 472
16.2 Configuration 473
16.3 Pins 475
16.4 Registers 476
16.4.1 External Interrupt Request Level Registers (ELVR0 to ELVR3) 477
16.4.2 External Interrupt Request Registers (EIRR0 to EIRR3) 480
16.4.3 Enable Interrupt Request Registers (ENIR0 to ENIR3) 482
16.5 Explanation of Operations and Setting Procedure Examples 484
16.5.1 Operations of the External Interrupt Controllers 484
16.5.2 Return from Standby Mode 487
16.5.3 Return from Sleep Mode 489

CHAPTER 17 Watchdog Timer 491
17.1 Overview 492
17.2 Configuration 493
17.3 Registers 495
17.3.1 Watchdog Timer Control Register 0 (WDTCR0) 496
17.3.2 Watchdog Timer Clear Pattern Register 0 (WDTCPR0) 499
17.4 Explanation of Operations and Setting Procedure Examples 500
17.4.1 Operations of the Watchdog Timer 500

CHAPTER 18 Watch Counter 503
18.1 Overview 504
18.2 Configuration 505
18.3 Registers 507
18.3.1 Watch Counter Reload Register (WCRL) 508
18.3.2 Watch Counter Control Register (WCCR) 509
18.3.3 Watch Counter Read Register (WCRD) 512
18.4 Interrupts 513
18.5 Explanation of Operations and Setting Procedure Examples 514
18.5.1 Operations of the Watch Counter 514
18.6 Notes on Use 516

CHAPTER 19 32-bit Free-Run Timer 517
19.1 Overview 518
19.2 Configuration 519
19.3 Pins 523
19.4 Registers 524

19.4.1 Free-Run Timer Select Register (FRTSEL) 525
19.4.2 Compare Clear Register (CPCLR0, CPCLR1) 526
19.4.3 Timer Data Register (TCDT0, TCDT1) 527
19.4.4 Timer Status Control Register Upper/Lower (TCCSH0/TCCSL0, TCCSH1/TCCSL1) 528
19.5 Interrupts 532
19.6 An Explanation of Operations and Setting Procedure Examples 533
19.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected 534
19.6.2 Operation When an External Clock Is Selected 535

CHAPTER 20 32-bit Input Capture 537
20.1 Overview 538
20.2 Configuration 539
20.3 Pins 541
20.4 Registers 542
20.4.1 Input Capture Status Control Registers (ICS01 to ICS67) 543
20.4.2 Input Capture Data Register (IPCP0 to IPCP7) 547
20.5 Interrupts 548
20.6 An Explanation of Operations and Setting Procedure Examples 549
20.6.1 Explanation of 32-bit Input Capture Operation 549

CHAPTER 21 32-bit Output Compare 553
21.1 Overview 554
21.2 Configuration 555
21.3 Pins 557
21.4 Registers 558
21.4.1 Output Compare Register (OCCP0 to OCCP7) 559
21.4.2 Compare Control Register Upper (OCSH1, OCSH3, OCSH5, OCSH7) 560
21.4.3 Compare Control Register Lower (OCSL0, OCSL2, OCSL4, OCSL6) 564
21.5 Interrupts 567
21.6 An Explanation of Operations and Setting Procedure Examples 568
21.6.1 When the 2 Channels Are Used Independently of Each Other 568
21.6.2 When the 2 Channels Are Used as a Pair 570

CHAPTER 22 16-bit Reload Timer 573
22.1 Overview 574
22.2 Configuration 575
22.3 Pins 577
22.4 Registers 578
22.4.1 Timer Control Status Register (TMCSR0 to TMCSR2) 579
22.4.2 16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2) 585
22.4.3 16-bit Timer Register (TMR0 to TMR2) 586
22.5 Interrupts 587
22.6 An Explanation of Operations and Setting Procedure Examples 588
22.6.1 Operation in Interval Timer Mode 589
22.6.2 Operations in Event Counter Mode 602
22.6.3 Operation in Cascade Mode 607
22.7 Notes on Use 609

CHAPTER 23 Base Timer I/O Select Function 611
23.1 Overview 612
23.2 Configuration 613
23.3 Pin 614
23.4 Registers 616
23.4.1 Base Timer IO Select Register for Ch.0/1/2/3 (BTSEL0123) 617
23.4.2 Base Timer IO Select Register for Ch.4/5/6/7 (BTSEL4567) 620
23.4.3 Base Timer IO Select Register for Ch.8/9/A/B (BTSEL89AB) 623
23.4.4 Base Timer IO Select Register for Ch.C/D/E/F (BTSELCDEF) 626
23.4.5 Base Timer Same Time Soft Start Register (BTSSSR) 629
23.5 I/O Mode 634
23.5.1 I/O Mode 0 (16-bit Timer Standard Mode) 634
23.5.2 I/O Mode 1 (Timer Full Mode) 636
23.5.3 I/O Mode 2 (External Trigger Shared Mode) 638
23.5.4 I/O Mode 3 (Other Channel Trigger Shared Mode) 640
23.5.5 Operations in I/O Mode 4 (Timer Activation/Stop Mode) 642
23.5.6 Operations in I/O Mode 5 (Same Time Software Activation Mode) 645
23.5.7 Operations in I/O Mode 6 (Software Activation Timer Activation/Stop Mode) 647
23.5.8 Operations in I/O Mode 7 (Timer Activation Mode) 649
23.5.9 Operations in I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode) 651

CHAPTER 24 Base Timer 653
24.1 Overview of the Base Timer 654
24.2 Block Diagrams of the Base Timer 656
24.3 Base Timer's Registers 664
24.4 Operations of the Base Timer 675
24.5 32-bit Mode Operations 677
24.6 Notes of Using the Base Timer 679
24.7 Base Timer Interrupts 681
24.8 Base Timer Description by Function Mode 682
24.8.1 PWM Function 683
24.8.2 PPG Function 697
24.8.3 Reload Timer Function 712
24.8.4 PWC Function 724

CHAPTER 25 Up/Down Counter 739
25.1 Overview 740
25.2 Configuration 742
25.3 Pin 744
25.4 Registers 745
25.4.1 Reload Compare Register (RCR0 to RCR3) 746
25.4.2 Up-Down Count Register (UDCR0 to UDCR3) 748
25.4.3 Counter Control Register (CCR0 to CCR3) 749
25.4.4 Counter Status Register (CSR0 to CSR3) 754
25.5 Interrupt 757
25.6 An Explanation of Operations and Setting Procedure Examples 759
25.6.1 Operation in Timer Mode 763
25.6.2 Operations in Up/Down Count Mode 765
25.6.3 Operations in Phase Difference Count Mode (Multiplied by 2) 768

25.6.4 Operations in Phase Difference Count Mode (Multiplied by 4) 770

CHAPTER 26 10-Bit A/D Converter 773
26.1 Overview 774
26.2 Configuration 775
26.3 Pins 777
26.4 Registers 779
26.4.1 A/DC Control Registers (ADCR0) 780
26.4.2 A/DC Status Registers (ADSR0) 783
26.4.3 Scan Conversion Control Registers (SCCR0) 787
26.4.4 Scan Conversion FIFO Number Setting Register (SFNS0) 791
26.4.5 Scan Conversion FIFO Data Registers (SCFD0) 793
26.4.6 Scan Conversion Input Select Registers (SCIS20 to SCIS00) 796
26.4.7 Priority Conversion Control Registers (PCCR0) 798
26.4.8 Priority Conversion FIFO Number Setting Registers (PFNS0) 802
26.4.9 Priority Conversion FIFO Data Registers (PCFD0) 804
26.4.10 Priority Conversion Input Select Registers (PCIS0) 808
26.4.11 A/D Comparison Data Setting Registers (CMPD0) 811
26.4.12 A/D Comparison Control Registers (CMPCR0) 812
26.4.13 Sampling Time Setting Registers (ADST00, ADST10) 816
26.4.14 Sampling Time Select Registers (ADSS20 to ADSS00) 819
26.4.15 Compare Time Setting Registers (ADCT0) 820
26.5 Interrupts 822
26.6 Explanation of Operations and Setting Procedure Examples 824
26.6.1 Operation of A/D Scan Conversion 834
26.6.2 Operation of A/D Priority Conversion 837
26.6.3 FIFO Operations 840
26.6.4 Activating the DMA Controller (DMAC) 846

CHAPTER 27 8-bit D/A Converter 849
27.1 Overview 850
27.2 Configuration 851
27.3 Pins 853
27.4 Registers 854
27.4.1 D/A Data Registers (DADR0 to DADR2) 855
27.4.2 D/A Control Registers (DACR0 to DACR2) 856
27.5 Explanation of Operations and Setting Procedure Examples 857
27.5.1 Operations of the 8-bit D/A Converter 857

CHAPTER 28 Multi-function Serial Interface 859
28.1 Characteristics of Multi-function Serial Interface 860
28.2 UART (Asynchronous Serial Interface) 861
28.3 Overview of UART (Asynchronous Serial Interface) 862
28.4 Registers of UART (Asynchronous Serial Interface) 864
28.4.1 Serial Control Register (SCR) 869
28.4.2 Serial Mode Register (SMR) 871
28.4.3 Serial Status Register (SSR) 874
28.4.4 Extended Serial Control Register (ESCR) 877
28.4.5 Reception Data Register / Transmission Data Register (RDR/TDR) 879

28.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 882
28.4.7 FIFO Control Register 1 (FCR1) 884
28.4.8 FIFO Control Register 0 (FCR0) 886
28.4.9 FIFO Byte Register (FBYTE1/FBYTE2) 889
28.5 Interrupts of UART 891
28.5.1 Occurrence of Reception Interrupts and Flag Set Timing 893
28.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing 894
28.5.3 Occurrence of Transmission Interrupts and Flag Set Timing 895
28.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing 896
28.6 Operation of UART 897
28.7 Dedicated Baud Rate Generator 902
28.7.1 Setting Baud Rate 903
28.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode) 907
28.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode) . 909
28.10 CSIO (Clock Synchronous Serial Interface) 913
28.11 Overview of CSIO (Clock Synchronous Serial Interface) 914
28.12 Registers of CSIO (Clock Synchronous Serial Interface) 915
28.12.1 Serial Control Register (SCR) 921
28.12.2 Serial Mode Register (SMR) 924
28.12.3 Serial Status Register (SSR) 927
28.12.4 Extended Serial Control Register (ESCR) 929
28.12.5 Reception Data Register / Transmission Data Register (RDR/TDR) 931
28.12.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 934
28.12.7 FIFO Control Register 1 (FCR1) 935
28.12.8 FIFO Control Register 0 (FCR0) 937
28.12.9 FIFO Byte Register (FBYTE1/FBYTE2) 940
28.12.10Serial Mode Select Registers (SSEL0123, SSEL4567) 942
28.12.11Received Data Mirror Registers/Transmitted Data Mirror Registers (RDRM/TDRM) 944
28.13 Interrupts of CSIO (Clock Synchronous Serial Interface) 945
28.13.1 Occurrence of Reception Interrupts and Flag Set Timing 947
28.13.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing 948
28.13.3 Occurrence of Transmission Interrupts and Flag Set Timing 950
28.13.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing 951
28.14 Operation of CSIO (Clock Synchronous Serial Interface) 952
28.15 Dedicated Baud Rate Generator 971
28.15.1 Setting Baud Rate 972
28.16 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface) 975
28.17 I^2C Interface 977
28.18 Overview of I^2C Interface 978
28.19 Registers of I^2C Interface 979
28.19.1 I^2C Bus Control Register (IBCR) 985
28.19.2 Serial Mode Register (SMR) 990
28.19.3 I^2C Bus Status Register (IBSR) 992
28.19.4 Serial Status Register (SSR) 996
28.19.5 Reception Data Register / Transmission Data Register (RDR/TDR) 998
28.19.6 7-bit Slave Address Mask Register (ISMK) 1000
28.19.7 7-bit Slave Address Register (ISBA) 1001
28.19.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0) 1002

28.19.9 FIFO Control Register 1 (FCR1) 1003
28.19.10 FIFO Control Register 0 (FCR0) 1005
28.19.11 FIFO Byte Register (FBYTE1/FBYTE2) 1009
28.20 Interrupts of I2C Interface 1011
28.20.1 Operation of I2C Interface Communication 1013
28.20.2 Master Mode 1014
28.20.3 Slave Mode 1031
28.20.4 Bus Error 1035
28.21 Dedicated Baud Rate Generator 1036
28.21.1 Example of I2C Flowcharts 1038

CHAPTER 29 USB Clock Generating Part 1051
29.1 Overview 1052
29.2 Configuration 1053
29.3 Registers 1054
29.3.1 USB Clock Configuration Register (UCCR) 1055
29.4 Explanation of Operations and Setting Procedure Examples 1057

CHAPTER 30 DMA Transfer Request Selector 1059
30.1 Overview 1060
30.2 Configuration 1061
30.3 Registers 1062
30.3.1 DREQ Select Register (DREQSEL) 1063

CHAPTER 31 USB Function 1065
31.1 Overview 1066
31.2 Configuration 1067
31.3 Registers 1068
31.3.1 USB Selection Register (USBSEL) 1071
31.3.2 USB Enable Register (USBEN) 1072
31.3.3 UDC Control Register (UDCC) 1073
31.3.4 EP0 Control Register (EP0C) 1077
31.3.5 EP1 to 5 Control Registers (EP1C to EP5C) 1079
31.3.6 Time Stamp Register (TMSP) 1084
31.3.7 UDC Status Register (UDCS) 1085
31.3.8 UDC Enable Interrupt Request Register (UDCIE) 1089
31.3.9 EP0I Status Register (EP0IS) 1091
31.3.10 EP0O Status Register (EP0OS) 1093
31.3.11 EP1 to 5 Status Registers (EP1S to EP5S) 1096
31.3.12 EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) 1100
31.4 Explanation of Operations and Setting Procedure Examples 1102
31.4.1 Detection of Connections and Disconnections 1105
31.4.2 Register Operation for a Command Response 1108
31.4.3 STALL Response and Release 1110
31.4.4 Suspend Function 1114
31.4.5 Wakeup Function 1115
31.4.6 DMA Transfer Function 1117
31.4.7 NULL Transfer Function 1122
31.4.8 Software Control Examples 1123

CHAPTER 32 USB Mini-HOST 1129
32.1 Overview 1130
32.2 Configuration 1132
32.3 Registers 1134
32.3.1 Host Control Register 0, 1 (HCNT) 1136
32.3.2 Host Interrupt Register (HIRQ) 1141
32.3.3 Host Error Status Register (HERR) 1145
32.3.4 Host Status Register (HSTATE) 1149
32.3.5 SOF Interrupt FRAME comparison register (HFCOMP) 1152
32.3.6 Retry Timer Setting Register (HRTIMER) 1153
32.3.7 Host Address Register (HADR) 1155
32.3.8 EOF Setting Register (HEOF) 1156
32.3.9 FRAME Setting Register (HFRAME) 1158
32.3.10 Host Token Endpoint Register (HTOKEN) 1160
32.4 Explanation of Operations and Setting Procedure Examples 1162
32.4.1 Connecting a USB Device 1162
32.4.2 Resetting the USB Bus 1164
32.4.3 Token Packets 1165
32.4.4 Data Packets 1166
32.4.5 Operation of a Handshake Packet 1167
32.4.6 Retry Function 1167
32.4.7 SOF Interrupts 1169
32.4.8 Error Status 1171
32.4.9 Packet End 1172
32.4.10 Suspend and Resume 1173
32.4.11 Disconnecting a USB Device 1175
32.5 Flowcharts for tokens in operation with USB Mini-HOST 1176

CHAPTER 33 DMA Controller (DMAC) 1181
33.1 Overview 1182
33.2 Configuration 1185
33.3 Pins 1187
33.4 Registers 1188
33.4.1 DMA Control Register (DMACR) 1190
33.4.2 DMA Source Address Registers (DSAR0 to DSAR7) 1193
33.4.3 DMA Destination Address Registers (DDAR0 to DDAR7) 1194
33.4.4 DMA Transfer Count Registers (DTCR0 to DTCR7) 1195
33.4.5 DMA Channel Control Registers (DCCR0 to DCCR7) 1196
33.4.6 DMA Channel Status Registers (DCSR0 to DCSR7) 1205
33.4.7 DMA-Halt by Interrupt Level Register (DILVR) 1208
33.5 Interrupts 1210
33.6 An Explanation of Operations and Setting Procedure Examples 1211
33.6.1 Transfer Settings 1211
33.6.2 Transfer Operations 1214
33.6.3 Transfer Suspension 1224
33.6.4 Operation at the End of Transfer 1226
33.6.5 Post-transfer Operation 1227
33.6.6 DMA Transfer Halt 1231

CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function 1233
34.1 Overview 1234
34.2 Configuration 1235
34.3 Registers 1237
34.3.1 IO-Data Request Registers (IORR0 to IORR7) 1239
34.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0) 1245
34.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1) 1247
34.3.4 Select Register 2 for DMA Transfer Request Clear by a Peripheral Function (ICSEL2) 1249
34.3.5 Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3) 1251
34.3.6 Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4) 1253
34.3.7 Select Register 5 for DMA Transfer Request Clear by a Peripheral Function (ICSEL5) 1255
34.3.8 Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6) 1258
34.3.9 Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7) 1260
34.3.10 Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8) 1262
34.3.11 Select Register 9 for DMA Transfer Request Clear by a Peripheral Function (ICSEL9) 1264
34.3.12 Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10) 1266
34.3.13 Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11) 1270
34.3.14 Select Register 12 for DMA Transfer Request Clear by a Peripheral Function (ICSEL12) 1273
34.3.15 Select Register 13 for DMA Transfer Request Clear by a Peripheral Function (ICSEL13) 1276
34.3.16 Select Register 14 for DMA Transfer Request Clear by a Peripheral Function (ICSEL14) 1279
34.4 An Explanation of Operations and Setting Procedure Examples 1281
34.4.1 Operations upon a DMA Transfer 1281

CHAPTER 35 Control of Built-in Program Memory 1283
35.1 Overview of Built-in Program Memory Controller 1284
35.2 Register for Built-in Program Memory Controller 1285
35.2.1 FLASH Control Register (FCTLR) 1286

CHAPTER 36 Flash Memory 1289
36.1 Overview of Flash Memory 1290
36.2 Flash Memory Configuration 1291
36.3 Flash Memory Registers 1293
36.3.1 FLASH Status Register (FSTR) 1294
36.3.2 FLASH Control Register (FCTLR) 1295
36.4 Flash Memory Access Mode 1296
36.5 Automatic Programming Algorithm 1297
36.5.1 Command Sequence 1297
36.5.2 Execution State of Automatic Programming Algorithm 1300
36.6 Explanation of Flash Memory Operation 1305
36.6.1 Read/Reset Operation 1305
36.6.2 Write Operation 1306
36.6.3 Chip Erase 1309
36.6.4 Sector Erase 1309
36.6.5 Sector Erase Suspending 1311
36.6.6 Sector Erase Restarting 1313
36.7 Notes on Using Flash Memory 1314

CHAPTER 37 Wild Register 1315
37.1 Overview of Wild Register 1316
37.2 Configuration of Wild Register 1317
37.3 Registers of Wild Register 1318
37.3.1 Wild Register Address Register (WRAR00 to WRAR15) 1320
37.3.2 Wild Register Data Register (WRDR00 to WRDR15) 1321
37.3.3 Wild Register Enable Register (WREN) 1322
37.4 Explanation of Operations and Setting Procedure Examples of the Wild Register 1323
37.4.1 Wild Register Operation 1323
37.5 Notes on Using the Wild Register 1324

CHAPTER 38 Serial Programming Connection 1327
38.1 Fujitsu Microelectronics Serial Programmer 1328
38.1.1 Pins Used 1335

CHAPTER 39 Handling the Device 1337
39.1 Notes on Handling the Device 1338

APPENDIX A I/O Map 1346

APPENDIX B List of Registers 1370

APPENDIX C Interrupt Vectors 1397

APPENDIX D Pin State in Each CPU State 1401

APPENDIX E Lists of Instructions 1415
E.1 Instruction List 1415
E.2 Instruction Tables 1419
E.3 List of Instructions That Can Be Specified for Delay Slots 1432

Index of Pins 1435

Main changes in this edition

Page	Changes (For details, refer to main body.)
-	First edition

CHAPTER 1 Overview

This chapter explains the features and basic specifications of the MB91660 series.

1.1 MB91660 Series Overview
1.2 MB91660 Series Product Configuration
1.3 MB91660 Series Block Diagram
1.4 Package Dimensions

1.1 MB91660 Series Overview

The MB91660 series, a microcontroller that uses 32-bit RISC CPUs, has built-in peripheral control functions for embedded control which requires high-performance/high-speed CPU processing. This series is based on the FR80 family CPUs and is implemented in a single-chip.

■ FR80 family CPUs

- 32-bit RISC, load/store architecture, 5-stage pipeline
- 16 general-purpose 32-bit registers
- 16-bit fixed-length instructions (basic instructions), 1 instruction per cycle
- Instructions suitable for embedded applications
  - Instructions for memory-to-memory transfer, bit processing, barrel shift, etc.
  - High-level language support instructions
    Function entry/exit instructions and multi-load/store instructions for register contents
  - Bit search instruction
    1 detection, 0 detection, and transition point detection
  - Branch instruction with delay slot(s)
    Reduced overhead time in branch executions
  - Register interlock function
    Efficient assembly language coding
  - Support for multipliers at the built-in function/instruction level
    Signed 32-bit multiplication - 5 cycles
    Signed 16-bit multiplication - 3 cycles
  - Interrupt (Save PC and PS)
    High-speed response at a minimum of 6 cycles, 16 levels of priority
  - Simultaneous access to a program and data enabled by Harvard architecture
    The prefetch function for instructions using the 4-word instruction queue in the CPU
- Basic instruction compatibility with the FR family CPUs
  - Addition of the bit search instruction
  - No resource instruction and coprocessor instruction provided

■ Maximum operating frequency

CPU	33 MHz
Peripheral	33 MHz
External bus	33 MHz

■ External bus interface

- Maximum operating frequency: 33 MHz
- 24 addresses, 8/16-bit data I/O (split bus/multiplex bus)
- Chip select output support for 4 separate areas that can be specified

  A programmable auto wait cycle for each area occurs.

■ DMA controller (DMAC)

- Number of channels: 8
- Address space: 32 bits (4 GB)
- Transfer mode: Block transfer/Burst transfer/Demand transfer
- Address update: Increment/Decrement/Fix (increment/decrement value fixed to 1, 2, or 4)
- Transfer size: 8 bits, 16 bits, and 32 bits
- Block size: 1 to 16
- Transfer count: 1 to 65,535 times
- Transfer request:
  - Request by software
  - Interrupt request of a built-in peripheral function (a shared interrupt request or external interrupt request)
  - Request by an external pin
- Reload function: Reloading of all channels can be specified.
- Level of priority: Fixed (ch.0 > ch.1 > ch.2 > ch.3 > ...), or round robin
- Interrupt request: Occurrence of a normal end interrupt request, abnormal end interrupt request, or transfer suspension interrupt request

■ Multifunction serial interface

- 4 channels with 16-byte FIFO, 8 channels without FIFO
- Any of the following uses can be selected for each channel (For ch.0, I^2C is not available.):
  - UART
  - CSIO
  - I^2C

[Features of UART]

- Full-duplex double buffer
- Selection with or without parity supported
- Built-in dedicated baud rate generator
- External clock available as a serial clock
- Various error detection functions available (parity errors, framing errors, and overrun errors)

[Features of CSIO]

- Full-duplex double buffer
- Built-in dedicated baud rate generator
- Overrun error detection function available

[Features of I^2C]

- Standard mode (maximum: 100 kbps)/High-speed mode (maximum: 400 kbps) supported
- 5V tolerance supported for some channels

■ Interrupts

- Total of 32 external interrupts (5V tolerance supported for some pins)
- Interrupt from an internal peripheral function
- Programmable setting of interrupt levels (16 levels)
- Return from stop mode or sleep mode supported

■ A/D converter

- 24 channels, 1 unit
- 10-bit resolution
- Successive comparison type Conversion time: Approximately 1.2 µs (PCLK=33 MHz)
- Priority A/D conversion available (2 levels)
- Conversion mode (one shot conversion mode, scanning conversion mode)
- Activation trigger (software/external trigger/base timer)
- FIFO for storing conversion data available (scanning conversion: 16 levels; priority conversion: 4 levels)

■ D/A converter

- Number of channels: 3 built-in channels
- 8-bit resolution

■ Base timer

- Number of channels: 16 built-in channels
- Any of the following uses can be selected for each channel:
  - 16/32-bit reload timer
  - 16-bit PWM timer
  - 16/32-bit PWC timer
  - 16-bit PPG timer
- 32-bit timer available by connecting 2 channels in cascade
- Function for activating multiple channels simultaneously available
- I/O select function available

■ 16-bit reload timer

- Number of channels: 3 (including a channel for REALOS)
- Interval timer function
- Function for selecting count clock (Peripheral clock (PCLK) divided by a value ranging from 2 to 64)

■ Compare timer

- 32-bit input capture: 8 built-in channels
- 32-bit output compare: 8 built-in channels
- 32-bit free-run timer: 2 built-in channels

■ Other interval timers

- Up/Down counter: 4 built-in channels
- Watch counter: 1 built-in channel
- Watchdog timer: 1 built-in channel

■ USB function with Mini-HOST

- Number of channels: 1
- Supports Full-Speed only
- The USB function and USB Mini-HOST are the switch types (USB I/O multiplexed)
- Support of DMA transfer

[USB Function]

- Support of up to six endpoints
  - Endpoint 0 is provided for the fixed use of control transfers
  - Bulk or interrupt transfer can be selected for endpoint 1 to 5
- Double buffer structure for endpoint 1 to 5

[USB Mini-HOST]

- Support of bulk and interrupt transfer (Only using endpoint1 and endpoint2)
- Automatic detection of connection/disconnection of USB devices
- Automatic processing of a handshake packet for IN/OUT token processing
- Support of a maximum packet length of up to 256 bytes
- Support for a wakeup function

■ Slave interface

- Equipped with a register for communication along with large-capacity transmission/ reception FIFO that enables large amount of data transmission and reception.
- Buffer: reception 1.5 Kbytes, transmission 1.5 Kbytes
- Host function: A function to process by storing to a buffer. Once the buffer is full, this function stops data reception. Communication control is possible during data transmission/ reception. This enables CPU load reduction and efficiency of communication.
- 16-bit data port
- Transmission/ reception data port control function

■ Main timer

- Number of channels: 1
- Count of the oscillation stabilization wait time of the main clock (MCLK).
- Count of the oscillation stabilization wait time of the PLL clock (PLLCLK).
- Interval timer when the oscillation of the main clock (MCLK) is stable.

■ Sub timer

- Number of channels: 1
- Count of the oscillation stabilization wait time of the sub clock (SBCLK).
- Interval timer when the oscillation of the sub clock (SBCLK) is stable.

■ Clock generation

- Main clock (MCLK) oscillation
- Sub clock (SBCLK) oscillation
- PLL clock (PLLCLK) oscillation

■ Low-power dissipation mode

- Stop mode
- Watch mode
- Sleep mode
- Doze mode
- Clock division function

■ Other features

- I/O port
- $\overline{\text{INIT}}$ pin available as a reset pin.
- Watchdog timer reset and software reset available
- Delay interrupt
- Power supply
  - Single power supply (When USB not used: 2.7 V to 3.6 V, When USB used: 3.0 V to 3.6 V)

1.2 MB91660 Series Product Configuration

This section explains the products in the MB91660 series.

Table 1.2-1 MB91660 series product configuration (1 / 2)

Part number / Parameter	MB91V650	MB91F662	MB91661	MB91664
Product type	Evaluation product	Flash memory product	MASK ROM product	
Built-in program memory capacity	- (Support by emulation memory)	512 Kbytes (Flash)	384 Kbytes (ROM)	128 Kbytes (ROM)
Built-in RAM capacity	128 Kbytes	48 Kbytes	32 Kbytes	16 Kbytes
External bus interface	Yes			
DMA controller (DMAC)	8 channels			
Base timer	16 channels			
Multifunction serial interface	Without FIFO : 8 channels (ch.0 to ch.7) With FIFO : 4 channels (ch.8 to ch.11)			
External interrupt	32 (some pins are 5 V tolerant)			
10-bit A/D converter	32 channels (2 units)	24 channels (1 unit)		
8-bit D/A converter	3 channels			
16-bit reload timer	3 channels			
Compare timer	32-bit input capture : 8 channels 32-bit output compare : 8 channels 32-bit free-run timer : 2 channels			
Up/Down counter	4 channels			
Watch counter	1 channel			
I/O port	154	99		
USB function with Mini-HOST	1 channel			
Slave interface	Yes			
Main timer	1 channel			
Sub timer	1 channel			
Wild register	16 channels			

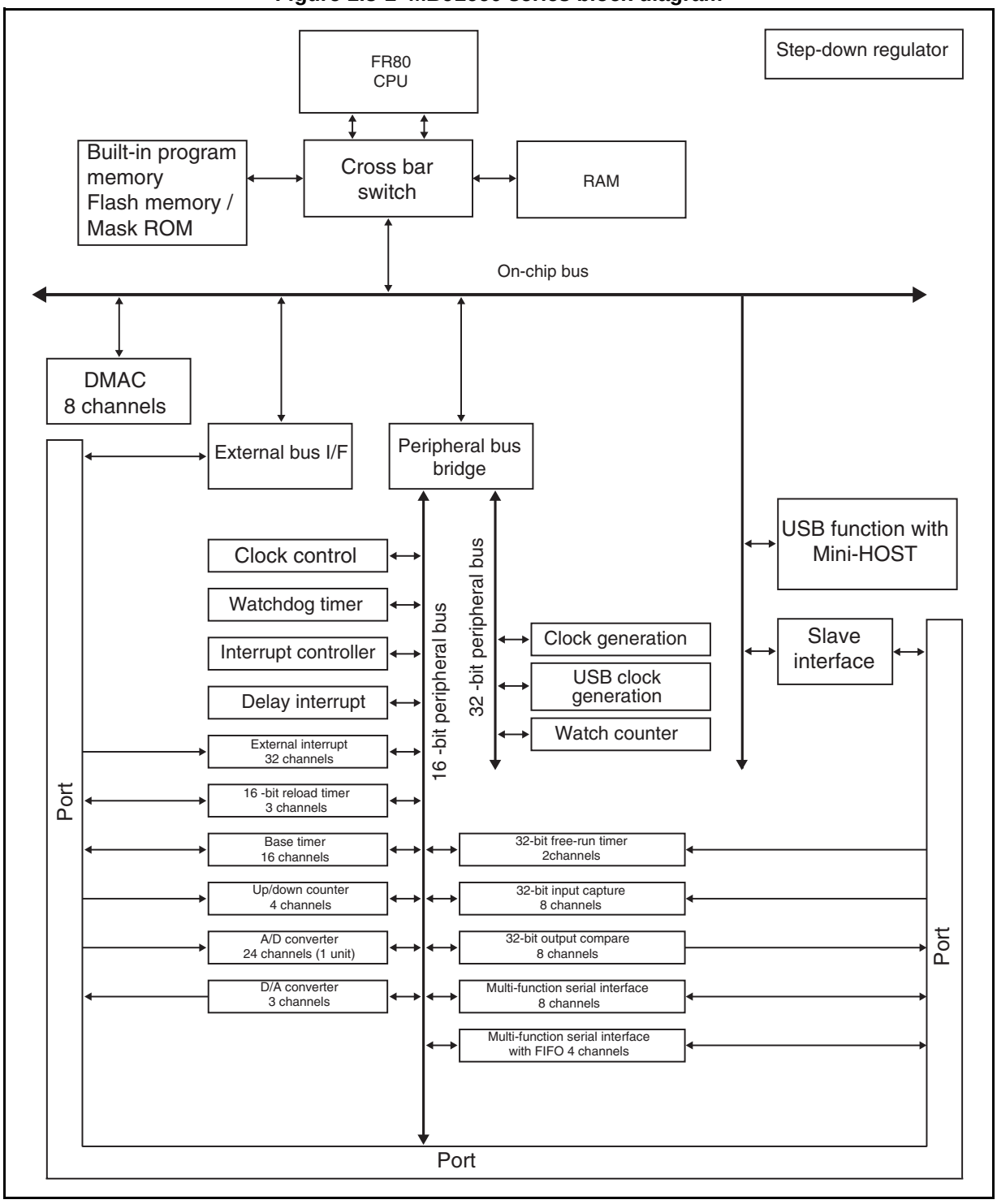
Table 1.2-1 MB91660 series product configuration (2 / 2)

Part number / Parameter	MB91V650	MB91F662	MB91661	MB91664
Debug function	DSU4	-		
Package	-	Type: LQFP-120 Package code: FPT-120P-M21 Pin pitch: 0.50mm Size:16.0mm × 16.0mm		

1.3 MB91660 Series Block Diagram

Figure 1.3-1 is the block diagrams of the MB91660 series.

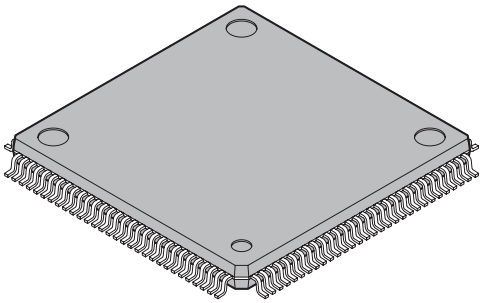
Figure 1.3-1 MB91660 series block diagram

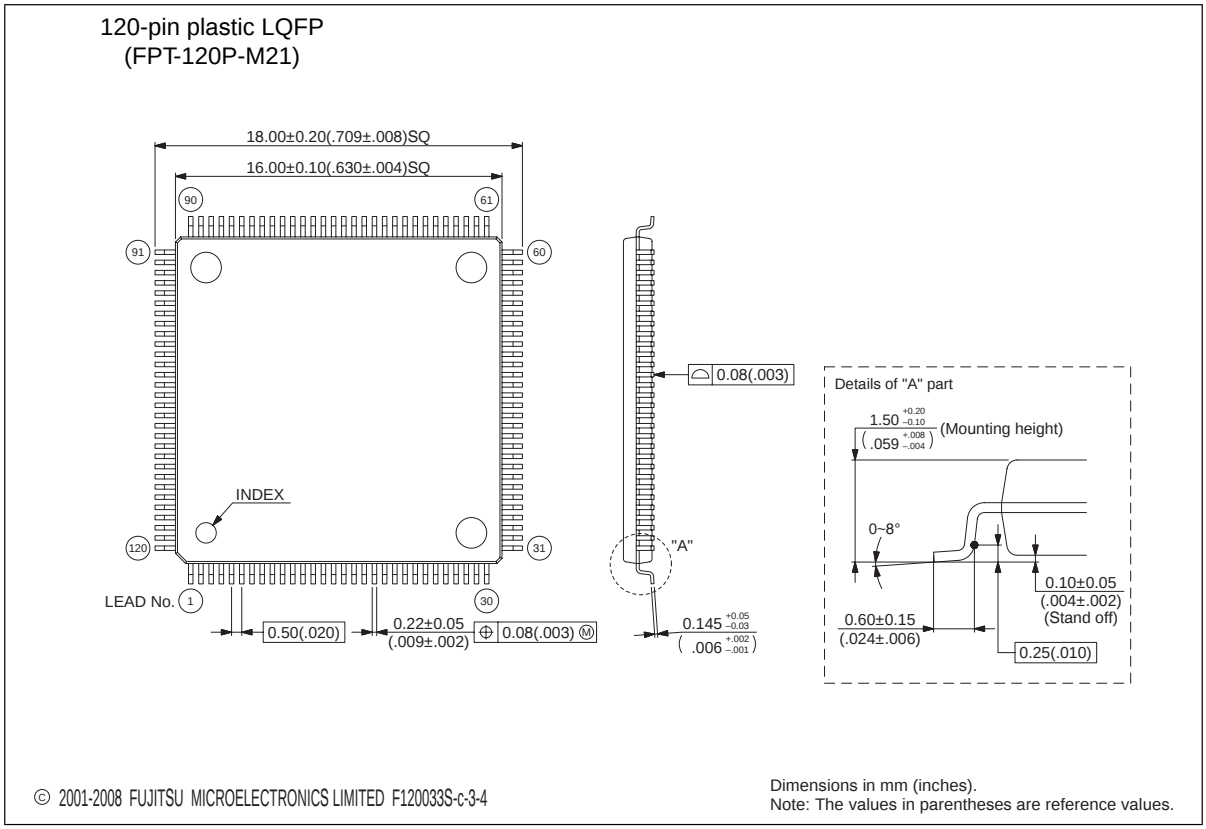


1.4 Package Dimensions

The dimensions of the packages used for the MB91660 series are shown below.

Figure 1.4-1 Package dimensions (FPT-120P-M21)

120-pin plastic LQFP		
	Lead pitch	0.50 mm
	Package width × package length	16.0 × 16.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.88 g
(FPT-120P-M21)		



For the latest specifications of package dimensions, refer to the following URL.
http://edevice.fujitsu.com/package/en-search/

CHAPTER 2 Pins of the MB91660 Series

This chapter explains the pins and multiplexed pin settings of the MB91660 series.

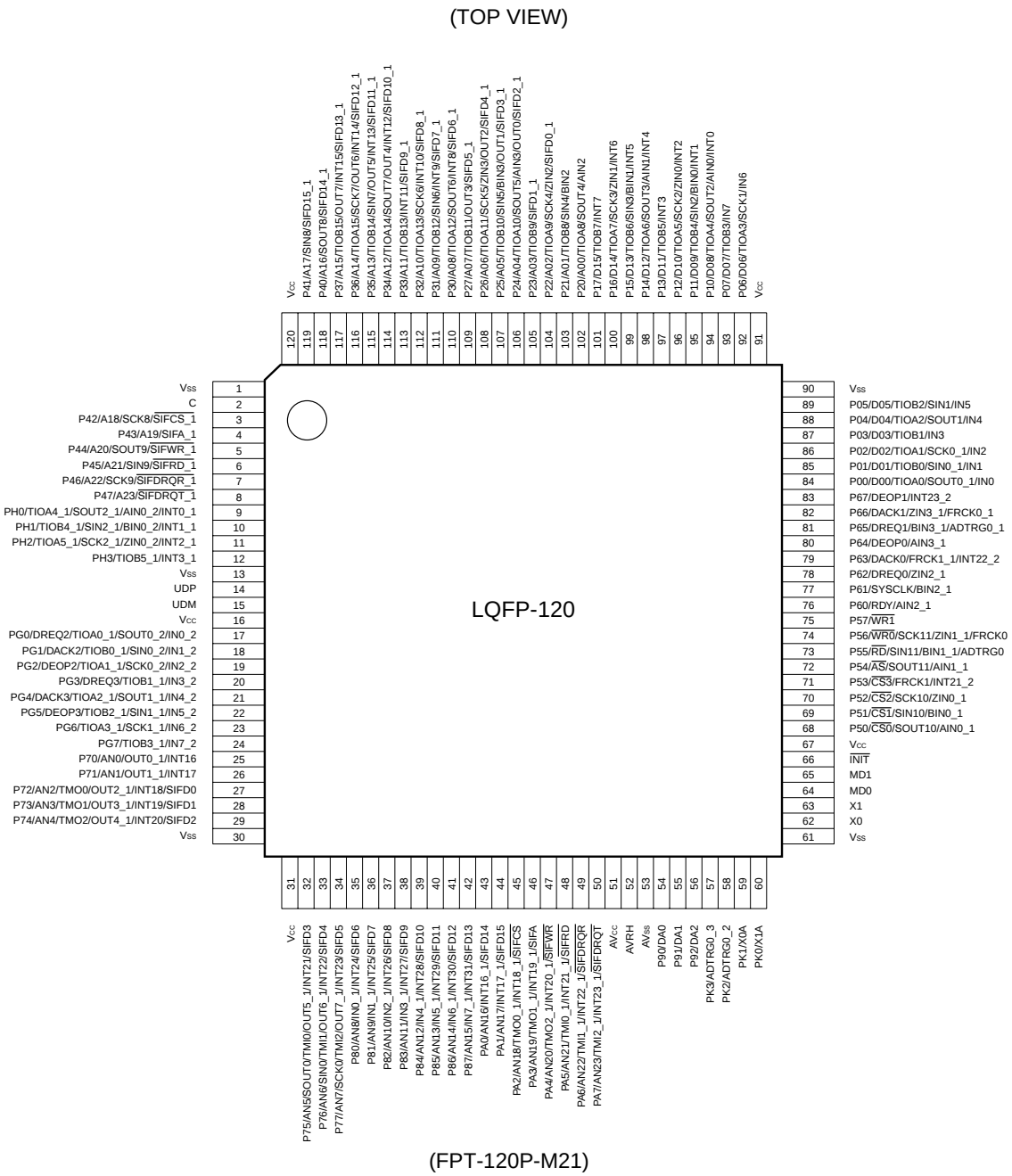
2.1 Pin Assignment Diagram
2.2 Pin Functions
2.3 I/O Circuit Types
2.4 Setting Method for Pins

2.1 Pin Assignment Diagram

1 type of package is available for the MB91660 series.

■ LQFP-120

Figure 2.1-1 Pin assignment diagram in the LQFP-120 series



Note : The number after the underscore (“_”) in pin names such as XXX_1 and XXX_2 indicates the port number. For these pins, there are multiple pins that provide the same function for the same channel. Use the extended port function register (EPFR) to select the pin.

2.2 Pin Functions

Table 2.2-1 lists the pin functions of the MB91660 series.
In a pin that includes an underscore (_), such as XXX_1 and XXX_2, the number following the underscore represents a port number. For details of the port numbers, see "2.4 Setting Method for Pins".

■ Pin function list

Table 2.2-1 Pin functions (1 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
1	V_{SS}	-	GND pin	-	-
2	C	-	Power stabilization capacity pin	-	-
3	P42	Q*2	General-purpose I/O port	-	❍
	A18		External bus interface address bus bit18	-	-
	SCK8 (SCL8)		Multifunction serial interface ch.8 clock I/O pin. This pin operates as SCK8 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL8 when it is used in an I^2C (operation mode 4).	-	❍
	$\overline{\text{SIFCS\_1}}$		Slave interface chip select signal input pin (Port 1)	❍	-
4	P43	Q*2	General-purpose I/O port	-	❍
	A19		External bus interface address bus bit19	-	-
	SIFA_1		Slave interface address signal input pin (Port 1)	❍	-
5	P44	Q*2	General-purpose I/O port	-	❍
	A20		External bus interface address bus bit20	-	-
	SOUT9 (SDA9)		Multifunction serial interface ch.9 output pin. This pin operates as SOUT9 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA9 when it is used in an I^2C (operation mode 4).	-	❍
	$\overline{\text{SIFWR\_1}}$		Slave interface write strobe signal input pin (Port 1)	❍	-

Table 2.2-1 Pin functions (2 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
6	P45	Q*2	General-purpose I/O port	-	❍
	A21		External bus interface address bus bit21	-	-
	SIN9		Multifunction serial interface ch.9 input pin	-	❍
	$\overline{\text{SIFRD\_1}}$		Slave interface read strobe signal input pin (Port 1)	❍	-
7	P46	D*2	General-purpose I/O port	-	❍
	A22		External bus interface address bus bit22	-	-
	SCK9 (SCL9)		Multifunction serial interface ch.9 clock I/O pin. This pin operates as SCK9 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL9 when it is used in an I^2C (operation mode 4).	-	❍
	$\overline{\text{SIFDRQR\_1}}$		Slave interface received data request signal output pin (Port 1)	-	-
8	P47	D*2	General-purpose I/O port	-	❍
	A23		External bus interface address bus bit23	-	-
	$\overline{\text{SIFDRQT\_1}}$		Slave interface transmitted data request signal output pin (Port 1)	-	-
9	PH0	D*2	General-purpose I/O port	-	❍
	TIOA4_1		Base timer ch.4 TIOA pin (Port 1)	-	❍
	SOUT2_1 (SDA2_1)		Multifunction serial interface ch.2 output pin (Port 1). This pin operates as SOUT2_1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA2_1 when it is used in an I^2C (operation mode 4).	-	❍
	AIN0_2		Up/Down counter ch.0 AIN input pin (Port 2)	-	❍
	INT0_1		External interrupt request 0 input pin (Port 1)	-	❍

Table 2.2-1 Pin functions (3 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
10	PH1	D*2	General-purpose I/O port	-	❍
	TIOB4_1		Base timer ch.4 TIOB pin (Port 1)	-	❍
	SIN2_1		Multifunction serial interface ch.2 input pin (Port 1)	-	❍
	BIN0_2		Up/Down counter ch.0 BIN input pin (Port 2)	-	❍
	INT1_1		External interrupt request 1 input pin (Port 1)	-	❍
11	PH2	D*2	General-purpose I/O port	-	❍
	TIOA5_1		Base timer ch.5 TIOA pin (Port 1)	-	❍
	SCK2_1 (SCL2_1)		Multifunction serial interface ch.2 clock I/O pin (Port 1). This pin operates as SCK2_1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL2_1 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN0_2		Up/Down counter ch.0 ZIN input pin (Port 2)	-	❍
	INT2_1		External interrupt request 2 input pin (Port 1)	-	❍
12	PH3	D*2	General-purpose I/O port	-	❍
	TIOB5_1		Base timer ch.5 TIOB pin (Port 1)	-	❍
	INT3_1		External interrupt request 3 input pin (Port 1)	-	❍
13	V_{SS}	-	GND pin	-	-
14	UDP	USB	D+ pin of USB function in Mini-HOST	-	-
15	UDM	USB	D- pin of USB function in Mini-HOST	-	-
16	V_{CC}	-	Power pin	-	-

Table 2.2-1 Pin functions (4 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
17	PG0	D*2	General-purpose I/O port	-	❍
	DREQ2		DMA controller (DMAC) ch.2 transfer request input pin	-	❍
	TIOA0_1		Base timer ch.0 TIOA pin (Port 1)	-	❍
	SOUT0_2		Multifunction serial interface ch.0 output pin (Port 2). This pin operates as SOUT0_2 when it is used in a UART/CSIO (operation modes 0 to 2).	-	-
	IN0_2		32-bit input capture ch.0 input pin (Port 2)	-	❍
18	PG1	D*2	General-purpose I/O port	-	❍
	DACK2		DMA controller (DMAC) ch.2 transfer request acceptance signal output pin	-	-
	TIOB0_1		Base timer ch.0 TIOB pin (Port 1)	-	❍
	SIN0_2		Multifunction serial interface ch.0 input pin (Port 2)	-	❍
	IN1_2		32-bit input capture ch.1 input pin (Port 2)	-	❍
19	PG2	D*2	General-purpose I/O port	-	❍
	DEOP2		DMA controller (DMAC) ch.2 last transfer signal output pin	-	-
	TIOA1_1		Base timer ch.1 TIOA pin (Port 1)	-	❍
	SCK0_2		Multifunction serial interface ch.0 clock I/O pin (Port 2). This pin operates as SCK0_2 when it is used in a UART/CSIO (operation modes 0 to 2).	-	❍
	IN2_2		32-bit input capture ch.2 input pin (Port 2)	-	❍
20	PG3	D*2	General-purpose I/O port	-	❍
	DREQ3		DMA controller (DMAC) ch.3 transfer request input pin	-	❍
	TIOB1_1		Base timer ch.1 TIOB pin (Port 1)	-	❍
	IN3_2		32-bit input capture ch.3 input pin (Port 2)	-	❍

Table 2.2-1 Pin functions (5 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
21	PG4	D*2	General-purpose I/O port	-	❍
	DACK3		DMA controller (DMAC) ch.3 transfer request acceptance signal output pin	-	-
	TIOA2_1		Base timer ch.2 TIOA pin (Port 1)	-	❍
	SOUT1_1 (SDA1_1)		Multifunction serial interface ch.1 output pin (Port 1). This pin operates as SOUT1_1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA1_1 when it is used in an I^2C (operation mode 4).	-	❍
	IN4_2		32-bit input capture ch.4 input pin (Port 2)	-	❍
22	PG5	D*2	General-purpose I/O port	-	❍
	DEOP3		DMA controller (DMAC) ch.3 last transfer signal output pin	-	-
	TIOB2_1		Base timer ch.2 TIOB pin (Port 1)	-	❍
	SIN1_1		Multifunction serial interface ch.1 input pin (Port 1)	-	❍
	IN5_2		32-bit input capture ch.5 input pin (Port 2)	-	❍
23	PG6	D*2	General-purpose I/O port	-	❍
	TIOA3_1		Base timer ch.3 TIOA pin (Port 1)	-	❍
	SCK1_1 (SCL1_1)		Multifunction serial interface ch.1 clock I/O pin (Port 1). This pin operates as SCK1_1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL1_1 when it is used in an I^2C (operation mode 4).	-	❍
	IN6_2		32-bit input capture ch.6 input pin (Port 2)	-	❍
24	PG7	D*2	General-purpose I/O port	-	❍
	TIOB3_1		Base timer ch.3 TIOB pin (Port 1)	-	❍
	IN7_2		32-bit input capture ch.7 input pin (Port 2)	-	❍

Table 2.2-1 Pin functions (6 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
25	P70	E	General-purpose I/O port	-	❍
	AN0		10-bit A/D converter ch.0 analog input pin	-	-
	OUT0_1		32-bit output compare ch.0 output pin (Port 1)	-	-
	INT16		External interrupt request 16 input pin	-	❍
26	P71	E	General-purpose I/O port	-	❍
	AN1		10-bit A/D converter ch.1 analog input pin	-	-
	OUT1_1		32-bit output compare ch.1 output pin (Port 1)	-	-
	INT17		External interrupt request 17 input pin	-	❍
27	P72	R	General-purpose I/O port	-	❍
	AN2		10-bit A/D converter ch.2 analog input pin	-	-
	TMO0		16-bit reload timer ch.0 output pin	-	-
	OUT2_1		32-bit output compare ch.2 output pin (Port 1)	-	-
	INT18		External interrupt request 18 input pin	-	❍
	SIFD0		Slave interface data I/O pin bit0	❍	-
28	P73	R	General-purpose I/O port	-	❍
	AN3		10-bit A/D converter ch.3 analog input pin	-	-
	TMO1		16-bit reload timer ch.1 output pin	-	-
	OUT3_1		32-bit output compare ch.3 output pin (Port 1)	-	-
	INT19		External interrupt request 19 input pin	-	❍
	SIFD1		Slave interface data I/O pin bit1	❍	-

Table 2.2-1 Pin functions (7 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
29	P74	R	General-purpose I/O port	-	❍
	AN4		10-bit A/D converter ch.4 analog input pin	-	-
	TMO2		16-bit reload timer ch.2 output pin	-	-
	OUT4_1		32-bit output compare ch.4 output pin (Port 1)	-	-
	INT20		External interrupt request 20 input pin	-	❍
	SIFD2		Slave interface data I/O pin bit2	❍	-
30	V_{SS}	-	GND pin	-	-
31	V_{CC}	-	Power pin	-	-
32	P75	R	General-purpose I/O port	-	❍
	AN5		10-bit A/D converter ch.5 analog input pin	-	-
	SOUT0		Multifunction serial interface ch.0 output pin. This pin operates as SOUT0 when it is used in a UART/CSIO (operation modes 0 to 2).	-	-
	TMI0		16-bit reload timer ch.0 input pin	-	❍
	OUT5_1		32-bit output compare ch.5 output pin (Port 1)	-	-
	INT21		External interrupt request 21 input pin	-	❍
	SIFD3		Slave interface data I/O pin bit3	❍	-
33	P76	R	General-purpose I/O port	-	❍
	AN6		10-bit A/D converter ch.6 analog input pin	-	-
	SIN0		Multifunction serial interface ch.0 input pin	-	❍
	TMI1		16-bit reload timer ch.1 input pin	-	❍
	OUT6_1		32-bit output compare ch.6 output pin (Port 1)	-	-
	INT22		External interrupt request 22 input pin	-	❍
	SIFD4		Slave interface data I/O pin bit4	❍	-

Table 2.2-1 Pin functions (8 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
34	P77	R	General-purpose I/O port	-	❍
	AN7		10-bit A/D converter ch.7 analog input pin	-	-
	SCK0		Multifunction serial interface ch.0 clock I/O pin This pin operates as SCK0 when it is used in a UART/CSIO (operation modes 0 to 2).	-	❍
	TMI2		16-bit reload timer ch.2 input pin	-	❍
	OUT7_1		32-bit output compare ch.7 output pin (Port 1)	-	-
	INT23		External interrupt request 23 input pin	-	❍
	SIFD5		Slave interface data I/O pin bit5	❍	-
35	P80	R	General-purpose I/O port	-	❍
	AN8		10-bit A/D converter ch.8 analog input pin	-	-
	IN0_1		32-bit input capture ch.0 input pin (Port 1)	-	❍
	INT24		External interrupt request 24 input pin	-	❍
	SIFD6		Slave interface data I/O pin bit6	❍	-
36	P81	R	General-purpose I/O port	-	❍
	AN9		10-bit A/D converter ch.9 analog input pin	-	-
	IN1_1		32-bit input capture ch.1 input pin (Port 1)	-	❍
	INT25		External interrupt request 25 input pin	-	❍
	SIFD7		Slave interface data I/O pin bit7	❍	-
37	P82	R	General-purpose I/O port	-	❍
	AN10		10-bit A/D converter ch.10 analog input pin	-	-
	IN2_1		32-bit input capture ch.2 input pin (Port 1)	-	❍
	INT26		External interrupt request 26 input pin	-	❍
	SIFD8		Slave interface data I/O pin bit8	❍	-

Table 2.2-1 Pin functions (9 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
38	P83	R	General-purpose I/O port	-	❍
	AN11		10-bit A/D converter ch.11 analog input pin	-	-
	IN3_1		32-bit input capture ch.3 input pin (Port 1)	-	❍
	INT27		External interrupt request 27 input pin	-	❍
	SIFD9		Slave interface data I/O pin bit9	❍	-
39	P84	R	General-purpose I/O port	-	❍
	AN12		10-bit A/D converter ch.12 analog input pin	-	-
	IN4_1		32-bit input capture ch.4 input pin (Port 1)	-	❍
	INT28		External interrupt request 28 input pin	-	❍
	SIFD10		Slave interface data I/O pin bit10	❍	-
40	P85	R	General-purpose I/O port	-	❍
	AN13		10-bit A/D converter ch.13 analog input pin	-	-
	IN5_1		32-bit input capture ch.5 input pin (Port 1)	-	❍
	INT29		External interrupt request 29 input pin	-	❍
	SIFD11		Slave interface data I/O pin bit11	❍	-
41	P86	R	General-purpose I/O port	-	❍
	AN14		10-bit A/D converter ch.14 analog input pin	-	-
	IN6_1		32-bit input capture ch.6 input pin (Port 1)	-	❍
	INT30		External interrupt request 30 input pin	-	❍
	SIFD12		Slave interface data I/O pin bit12	❍	-

Table 2.2-1 Pin functions (10 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
42	P87	R	General-purpose I/O port	-	❍
	AN15		10-bit A/D converter ch.15 analog input pin	-	-
	IN7_1		32-bit input capture ch.7 input pin (Port 1)	-	❍
	INT31		External interrupt request 31 input pin	-	❍
	SIFD13		Slave interface data I/O pin bit13	❍	-
43	PA0	R	General-purpose I/O port	-	❍
	AN16		10-bit A/D converter ch.16 analog input pin	-	-
	INT16_1		External interrupt request 16 input pin (Port 1)	-	❍
	SIFD14		Slave interface data I/O pin bit14	❍	-
44	PA1	R	General-purpose I/O port	-	❍
	AN17		10-bit A/D converter ch.17 analog input pin	-	-
	INT17_1		External interrupt request 17 input pin (Port 1)	-	❍
	SIFD15		Slave interface data I/O pin bit15	❍	-
45	PA2	R	General-purpose I/O port	-	❍
	AN18		10-bit A/D converter ch.18 analog input pin	-	-
	TMO0_1		16-bit reload timer ch.0 output pin (Port 1)	-	-
	INT18_1		External interrupt request 18 input pin (Port 1)	-	❍
	$\overline{\text{SIFCS}}$		Slave interface chip select signal input pin	❍	-

Table 2.2-1 Pin functions (11 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
46	PA3	R	General-purpose I/O port	-	❍
	AN19		10-bit A/D converter ch.19 analog input pin	-	-
	TMO1_1		16-bit reload timer ch.1 output pin (Port 1)	-	-
	INT19_1		External interrupt request 19 input pin (Port 1)	-	❍
	SIFA		Slave interface address signal input pin	❍	-
47	PA4	R	General-purpose I/O port	-	❍
	AN20		10-bit A/D converter ch.20 analog input pin	-	-
	TMO2_1		16-bit reload timer ch.2 output pin (Port 1)	-	-
	INT20_1		External interrupt request 20 input pin (Port 1)	-	❍
	$\overline{\text{SIFWR}}$		Slave interface write strobe signal input pin	❍	-
48	PA5	R	General-purpose I/O port	-	❍
	AN21		10-bit A/D converter ch.21 analog input pin	-	-
	TMI0_1		16-bit reload timer ch.0 input pin (Port 1)	-	❍
	INT21_1		External interrupt request 21 input pin (Port 1)	-	❍
	$\overline{\text{SIFRD}}$		Slave interface read strobe signal input pin	❍	-
49	PA6	E	General-purpose I/O port	-	❍
	AN22		10-bit A/D converter ch.22 analog input pin	-	-
	TMI1_1		16-bit reload timer ch.1 input pin (Port 1)	-	❍
	INT22_1		External interrupt request 22 input pin (Port 1)	-	❍
	$\overline{\text{SIFDRQR}}$		Slave interface received data request signal output pin	-	-

Table 2.2-1 Pin functions (12 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
50	PA7	E	General-purpose I/O port	-	❍
	AN23		10-bit A/D converter ch.23 analog input pin	-	-
	TMI2_1		16-bit reload timer ch.2 input pin (Port 1)	-	❍
	INT23_1		External interrupt request 23 input pin (Port 1)	-	❍
	$\overline{\text{SIFDRQT}}$		Slave interface transmitted data request signal output pin	-	-
51	AV_{CC}	-	10-bit A/D converter and 8-bit D/A converter analog power pin	-	-
52	AVRH	-	10-bit A/D converter analog reference voltage input pin	-	-
53	AV_{SS}	-	10-bit A/D converter and 8-bit D/A converter GND pin	-	-
54	P90	F	General-purpose I/O port	-	❍
	DA0		8-bit D/A converter ch.0 analog output pin	-	-
55	P91	F	General-purpose I/O port	-	❍
	DA1		8-bit D/A converter ch.1 analog output pin	-	-
56	P92	F	General-purpose I/O port	-	❍
	DA2		8-bit D/A converter ch.2 analog output pin	-	-
57	PK3	C	General-purpose I/O port	-	❍
	ADTRG0_3		10-bit A/D converter external trigger input pin (Port 3)	-	❍
58	PK2	C	General-purpose I/O port	-	❍
	ADTRG0_2		10-bit A/D converter external trigger input pin (Port 2)	-	❍
59	PK1	I	General-purpose I/O port	-	❍
	X0A		Sub clock (oscillation) input pin	-	❍
60	PK0	I	General-purpose I/O port	-	❍
	X1A		Sub clock (oscillation) I/O pin	-	-
61	V_{SS}	-	GND pin	-	-

Table 2.2-1 Pin functions (13 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
62	X0	A	Main clock (oscillation) input pin	-	❍
63	X1	A	Main clock (oscillation) I/O pin	-	-
64	MD0	H, P	Mode 0 pin. The I/O circuit type for the flash memory products is P. During normal operation, MD0=L must be input. During serial programming to flash memory, MD0=H must be input.	-	❍
65	MD1	H, P	Mode 1 pin. Input must always be at the "L" level. The I/O circuit type for the flash memory products is P.	-	❍
66	$\overline{INIT}$	H, P	External reset input pin. A reset is valid when $\overline{INIT}$=L. The I/O circuit type for the flash memory products is P.	-	❍
67	V_{CC}	-	Power pin	-	-
68	P50	C	General-purpose I/O port	-	❍
	$\overline{CS0}$		External bus interface chip select 0 output pin	-	-
	SOUT10 (SDA10)		Multifunction serial interface ch.10 output pin. This pin operates as SOUT10 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA10 when it is used in an I^2C (operation mode 4).	-	❍
	AIN0_1		Up/Down counter ch.0 AIN input pin (Port 1)	-	❍
69	P51	C	General-purpose I/O port	-	❍
	$\overline{CS1}$		External bus interface chip select 1 output pin	-	-
	SIN10		Multifunction serial interface ch.10 input pin	-	❍
	BIN0_1		Up/Down counter ch.0 BIN input pin (Port 1)	-	❍

Table 2.2-1 Pin functions (14 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
70	P52	C	General-purpose I/O port	-	❍
	$\overline{CS2}$		External bus interface chip select 2 output pin	-	-
	SCK10 (SCL10)		Multifunction serial interface ch.10 clock I/O pin. This pin operates as SCK10 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL10 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN0_1		Up/Down counter ch.0 ZIN input pin (Port 1)	-	❍
71	P53	C	General-purpose I/O port	-	❍
	$\overline{CS3}$		External bus interface chip select 3 output pin	-	-
	FRCK1		32-bit free-run timer ch.1 external clock input pin	-	❍
	INT21_2		External interrupt request 21 input pin (Port 2)	-	❍
72	P54	C	General-purpose I/O port	-	❍
	$\overline{AS}$		External bus interface address strobe output pin	-	-
	SOUT11 (SDA11)		Multifunction serial interface ch.11 output pin. This pin operates as SOUT11 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA11 when it is used in an I^2C (operation mode 4).	-	❍
	AIN1_1		Up/Down counter ch.1 AIN input pin (Port 1)	-	❍
73	P55	C	General-purpose I/O port	-	❍
	$\overline{RD}$		External bus interface read strobe output pin	-	-
	SIN11		Multifunction serial interface ch.11 input pin	-	❍
	BIN1_1		Up/Down counter ch.1 BIN input pin (Port 1)	-	❍
	ADTRG0		10-bit A/D converter external trigger input pin	-	❍

Table 2.2-1 Pin functions (15 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
74	P56	C	General-purpose I/O port	-	❍
	$\overline{WR0}$		External bus interface write strobe 0 output pin	-	-
	SCK11 (SCL11)		Multifunction serial interface ch.11 clock I/O pin. This pin operates as SCK11 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL11 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN1_1		Up/Down counter ch.1 ZIN input pin (Port 1)	-	❍
	FRCK0		32-bit free-run timer ch.0 external clock input pin	-	❍
75	P57	C	General-purpose I/O port	-	❍
	$\overline{WR1}$		External bus interface write strobe 1 output pin	-	-
76	P60	B	General-purpose I/O port	-	❍
	RDY		External bus interface ready input pin	❍	-
	AIN2_1		Up/Down counter ch.2 AIN input pin (Port 1)	-	❍
77	P61	C	General-purpose I/O port	-	❍
	SYSCLK		External bus interface bus clock output pin	-	-
	BIN2_1		Up/Down counter ch.2 BIN input pin (Port 1)	-	❍
78	P62	C	General-purpose I/O port	-	❍
	DREQ0		DMA controller (DMAC) ch.0 transfer request input pin	-	❍
	ZIN2_1		Up/Down counter ch.2 ZIN input pin (Port 1)	-	❍
79	P63	C	General-purpose I/O port	-	❍
	DACK0		DMA controller (DMAC) ch.0 transfer request acceptance signal output pin	-	-
	FRCK1_1		32-bit free-run timer ch.1 external clock input pin (Port 1)	-	❍
	INT22_2		External interrupt request 22 input pin (Port 2)	-	❍

Table 2.2-1 Pin functions (16 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
80	P64	C	General-purpose I/O port	-	❍
	DEOP0		DMA controller (DMAC) ch.0 last transfer signal output pin	-	-
	AIN3_1		Up/Down counter ch.3 AIN input pin (Port 1)	-	❍
81	P65	C	General-purpose I/O port	-	❍
	DREQ1		DMA controller (DMAC) ch.1 transfer request input pin	-	❍
	BIN3_1		Up/Down counter ch.3 BIN input pin (Port 1)	-	❍
	ADTRG0_1		10-bit A/D converter external trigger input pin (Port 1)	-	❍
82	P66	C	General-purpose I/O port	-	❍
	DACK1		DMA controller (DMAC) ch.1 transfer request acceptance signal output pin	-	-
	ZIN3_1		Up/Down counter ch.3 ZIN input pin (Port 1)	-	❍
	FRCK0_1		32-bit free-run timer ch.0 external clock input pin (Port 1)	-	❍
83	P67	C	General-purpose I/O port	-	❍
	DEOP1		DMA controller (DMAC) ch.1 last transfer signal output pin	-	-
	INT23_2		External interrupt request 23 input pin (Port 2)	-	❍
84	P00	B	General-purpose I/O port	-	❍
	D00		External bus interface data bus bit0	❍	-
	TIOA0		Base timer ch.0 TIOA pin	-	-
	SOUT0_1		Multifunction serial interface ch.0 output pin (Port 1). This pin operates as SOUT0_1 when it is used in a UART/CSIO (operation modes 0 to 2).	-	-
	IN0		32-bit input capture ch.0 input pin	-	❍

Table 2.2-1 Pin functions (17 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
85	P01	B	General-purpose I/O port	-	❍
	D01		External bus interface data bus bit1	❍	-
	TIOB0		Base timer ch.0 TIOB pin	-	❍
	SIN0_1		Multifunction serial interface ch.0 input pin (Port 1)	-	❍
	IN1		32-bit input capture ch.1 input pin	-	❍
86	P02	B	General-purpose I/O port	-	❍
	D02		External bus interface data bus bit2	❍	--
	TIOA1		Base timer ch.1 TIOA pin	-	❍
	SCK0_1		Multifunction serial interface ch.0 clock I/O pin (Port 1). This pin operates as SCK0_1 when it is used in a UART/CSIO (operation modes 0 to 2).	-	❍
	IN2		32-bit input capture ch.2 input pin	-	❍
87	P03	B	General-purpose I/O port	-	❍
	D03		External bus interface data bus bit3	❍	-
	TIOB1		Base timer ch.1 TIOB pin	-	❍
	IN3		32-bit input capture ch.3 input pin	-	❍
88	P04	B	General-purpose I/O port	-	❍
	D04		External bus interface data bus bit4	❍	-
	TIOA2		Base timer ch.2 TIOA pin	-	-
	SOUT1 (SDA1)		Multifunction serial interface ch.1 output pin. This pin operates as SOUT1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA1 when it is used in an I^2C (operation mode 4).	-	❍
	IN4		32-bit input capture ch.4 input pin	-	❍

Table 2.2-1 Pin functions (18 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
89	P05	B	General-purpose I/O port	-	❍
	D05		External bus interface data bus bit5	❍	-
	TIOB2		Base timer ch.2 TIOB pin	-	❍
	SIN1		Multifunction serial interface ch.1 input pin	-	❍
	IN5		32-bit input capture ch.5 input pin	-	❍
90	V_{SS}	-	GND pin	-	-
91	V_{CC}	-	Power pin	-	-
92	P06	B	General-purpose I/O port	-	❍
	D06		External bus interface data bus bit6	❍	-
	TIOA3		Base timer ch.3 TIOA pin	-	❍
	SCK1 (SCL1)		Multifunction serial interface ch.1 clock I/O pin. This pin operates as SCK1 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL1 when it is used in an I^2C (operation mode 4).	-	❍
	IN6		32-bit input capture ch.6 input pin	-	❍
93	P07	B	General-purpose I/O port	-	❍
	D07		External bus interface data bus bit7	❍	-
	TIOB3		Base timer ch.3 TIOB pin	-	❍
	IN7		32-bit input capture ch.7 input pin	-	❍
94	P10	B	General-purpose I/O port	-	❍
	D08		External bus interface data bus bit8	❍	-
	TIOA4		Base timer ch.4 TIOA pin	-	-
	SOUT2 (SDA2)		Multifunction serial interface ch.2 output pin. This pin operates as SOUT2 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA2 when it is used in an I^2C (operation mode 4).	-	❍
	AIN0		Up/Down counter ch.0 AIN input pin	-	❍
	INT0		External interrupt request 0 input pin	-	❍

Table 2.2-1 Pin functions (19 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
95	P11	B	General-purpose I/O port	-	❍
	D09		External bus interface data bus bit9	❍	-
	TIOB4		Base timer ch.4 TIOB pin	-	❍
	SIN2		Multifunction serial interface ch.2 input pin	-	❍
	BIN0		Up/Down counter ch.0 BIN input pin	-	❍
	INT1		External interrupt request 1 input pin	-	❍
96	P12	B	General-purpose I/O port	-	❍
	D10		External bus interface data bus bit10	❍	-
	TIOA5		Base timer ch.5 TIOA pin	-	❍
	SCK2 (SCL2)		Multifunction serial interface ch.2 clock I/O pin. This pin operates as SCK2 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL2 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN0		Up/Down counter ch.0 ZIN input pin	-	❍
	INT2		External interrupt request 2 input pin	-	❍
97	P13	B	General-purpose I/O port	-	❍
	D11		External bus interface data bus bit11	❍	-
	TIOB5		Base timer ch.5 TIOB pin	-	❍
	INT3		External interrupt request 3 input pin	-	❍
98	P14	B	General-purpose I/O port	-	❍
	D12		External bus interface data bus bit12	❍	-
	TIOA6		Base timer ch.6 TIOA pin	-	-
	SOUT3 (SDA3)		Multifunction serial interface ch.3 output pin. This pin operates as SOUT3 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA3 when it is used in an I^2C (operation mode 4).	-	❍
	AIN1		Up/Down counter ch.1 AIN input pin	-	❍
	INT4		External interrupt request 4 input pin	-	❍

Table 2.2-1 Pin functions (20 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
99	P15	B	General-purpose I/O port	-	❍
	D13		External bus interface data bus bit13	❍	-
	TIOB6		Base timer ch.6 TIOB pin	-	❍
	SIN3		Multifunction serial interface ch.3 input pin	-	❍
	BIN1		Up/Down counter ch.1 BIN input pin	-	❍
	INT5		External interrupt request 5 input pin	-	❍
100	P16	B	General-purpose I/O port	-	❍
	D14		External bus interface data bus bit14	❍	-
	TIOA7		Base timer ch.7 TIOA pin	-	❍
	SCK3 (SCL3)		Multifunction serial interface ch.3 clock I/O pin. This pin operates as SCK3 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL3 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN1		Up/Down counter ch.1 ZIN input pin	-	❍
	INT6		External interrupt request 6 input pin	-	❍
101	P17	B	General-purpose I/O port	-	❍
	D15		External bus interface data bus bit15	❍	-
	TIOB7		Base timer ch.7 TIOB pin	-	❍
	INT7		External interrupt request 7 input pin	-	❍
102	P20	D*2	General-purpose I/O port	-	❍
	A00		External bus interface address bus bit0	-	-
	TIOA8		Base timer ch.8 TIOA pin	-	-
	SOUT4 (SDA4)		Multifunction serial interface ch.4 output pin. This pin operates as SOUT4 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA4 when it is used in an I^2C (operation mode 4).	-	❍
	AIN2		Up/Down counter ch.2 AIN input pin	-	❍

Table 2.2-1 Pin functions (21 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
103	P21	D*2	General-purpose I/O port	-	❍
	A01		External bus interface address bus bit1	-	-
	TIOB8		Base timer ch.8 TIOB pin	-	❍
	SIN4		Multifunction serial interface ch.4 input pin	-	❍
	BIN2		Up/Down counter ch.2 BIN input pin	-	❍
104	P22	Q*2	General-purpose I/O port	-	❍
	A02		External bus interface address bus bit2	-	-
	TIOA9		Base timer ch.9 TIOA pin	-	❍
	SCK4 (SCL4)		Multifunction serial interface ch.4 clock I/O pin. This pin operates as SCK4 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL4 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN2		Up/Down counter ch.2 ZIN input pin	-	❍
	SIFD0_1		Slave interface data I/O pin bit0 (Port 1)	❍	-
105	P23	Q*2	General-purpose I/O port	-	❍
	A03		External bus interface address bus bit3	-	-
	TIOB9		Base timer ch.9 TIOB pin	-	❍
	SIFD1_1		Slave interface data I/O pin bit1 (Port 1)	❍	-
106	P24	Q*2	General-purpose I/O port	-	❍
	A04		External bus interface address bus bit4	-	-
	TIOA10		Base timer ch.10 TIOA pin	-	-
	SOUT5 (SDA5)		Multifunction serial interface ch.5 output pin. This pin operates as SOUT5 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA5 when it is used in an I^2C (operation mode 4).	-	❍
	AIN3		Up/Down counter ch.3 AIN input pin	-	❍
	OUT0		32-bit output compare ch.0 output pin	-	-
	SIFD2_1		Slave interface data I/O pin bit2 (Port 1)	❍	-

Table 2.2-1 Pin functions (22 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
107	P25	Q*2	General-purpose I/O port	-	❍
	A05		External bus interface address bus bit5	-	-
	TIOB10		Base timer ch.10 TIOB pin	-	❍
	SIN5		Multifunction serial interface ch.5 input pin	-	❍
	BIN3		Up/Down counter ch.3 BIN input pin	-	❍
	OUT1		32-bit output compare ch.1 output pin	-	-
	SIFD3_1		Slave interface data I/O pin bit3 (Port 1)	❍	-
108	P26	Q*2	General-purpose I/O port	-	❍
	A06		External bus interface address bus bit6	-	-
	TIOA11		Base timer ch.11 TIOA pin	-	❍
	SCK5 (SCL5)		Multifunction serial interface ch.5 clock I/O pin. This pin operates as SCK5 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL5 when it is used in an I^2C (operation mode 4).	-	❍
	ZIN3		Up/Down counter ch.3 ZIN input pin	-	❍
	OUT2		32-bit output compare ch.2 output pin	-	-
	SIFD4_1		Slave interface data I/O pin bit4 (Port 1)	❍	-
109	P27	Q*2	General-purpose I/O port	-	❍
	A07		External bus interface address bus bit7	-	-
	TIOB11		Base timer ch.11 TIOB pin	-	❍
	OUT3		32-bit output compare ch.3 output pin	-	-
	SIFD5_1		Slave interface data I/O pin bit5 (Port 1)	❍	-

Table 2.2-1 Pin functions (23 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
110	P30	Q*2	General-purpose I/O port	-	❍
	A08		External bus interface address bus bit8	-	-
	TIOA12		Base timer ch.12 TIOA pin	-	-
	SOUT6 (SDA6)		Multifunction serial interface ch.6 output pin. This pin operates as SOUT6 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA6 when it is used in an I^2C (operation mode 4).	-	❍
	INT8		External interrupt request 8 input pin	-	❍
	SIFD6_1		Slave interface data I/O pin bit6 (Port 1)	❍	-
111	P31	Q*2	General-purpose I/O port	-	❍
	A09		External bus interface address bus bit9	-	-
	TIOB12		Base timer ch.12 TIOB pin	-	❍
	SIN6		Multifunction serial interface ch.6 input pin	-	❍
	INT9		External interrupt request 9 input pin	-	❍
	SIFD7_1		Slave interface data I/O pin bit7 (Port 1)	❍	-
112	P32	Q*2	General-purpose I/O port	-	❍
	A10		External bus interface address bus bit10	-	-
	TIOA13		Base timer ch.13 TIOA pin	-	❍
	SCK6 (SCL6)		Multifunction serial interface ch.6 clock I/O pin. This pin operates as SCK6 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL6 when it is used in an I^2C (operation mode 4).	-	❍
	INT10		External interrupt request 10 input pin	-	❍
	SIFD8_1		Slave interface data I/O pin bit8 (Port 1)	❍	-
113	P33	Q*2	General-purpose I/O port	-	❍
	A11		External bus interface address bus bit11	-	-
	TIOB13		Base timer ch.13 TIOB pin	-	❍
	INT11		External interrupt request 11 input pin	-	❍
	SIFD9_1		Slave interface data I/O pin bit9 (Port 1)	❍	-

Table 2.2-1 Pin functions (24 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
114	P34	Q*2	General-purpose I/O port	-	❍
	A12		External bus interface address bus bit12	-	-
	TIOA14		Base timer ch.14 TIOA pin	-	-
	SOUT7 (SDA7)		Multifunction serial interface ch.7 output pin. This pin operates as SOUT7 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA7 when it is used in an I^2C (operation mode 4).	-	❍
	OUT4		32-bit output compare ch.4 output pin	-	-
	INT12		External interrupt request 12 input pin	-	❍
	SIFD10_1		Slave interface data I/O pin bit10 (Port 1)	❍	-
115	P35	Q*2	General-purpose I/O port	-	❍
	A13		External bus interface address bus bit13	-	-
	TIOB14		Base timer ch.14 TIOB pin	-	❍
	SIN7		Multifunction serial interface ch.7 input pin	-	❍
	OUT5		32-bit output compare ch.5 output pin	-	-
	INT13		External interrupt request 13 input pin	-	❍
	SIFD11_1		Slave interface data I/O pin bit11 (Port 1)	❍	-
116	P36	Q*2	General-purpose I/O port	-	❍
	A14		External bus interface address bus bit14	-	-
	TIOA15		Base timer ch.15 TIOA pin	-	❍
	SCK7 (SCL7)		Multifunction serial interface ch.7 clock I/O pin. This pin operates as SCK7 when it is used in a UART/CSIO (operation modes 0 to 2) and as SCL7 when it is used in an I^2C (operation mode 4).	-	❍
	OUT6		32-bit output compare ch.6 output pin	-	-
	INT14		External interrupt request 14 input pin	-	❍
	SIFD12_1		Slave interface data I/O pin bit12 (Port 1)	❍	-

Table 2.2-1 Pin functions (25 / 25)

Pin Number	Pin Name	I/O Circuit Type*1	Function	CMOS level input	CMOS level hysteresis input
117	P37	Q*2	General-purpose I/O port	-	❍
	A15		External bus interface address bus bit15	-	-
	TIOB15		Base timer ch.15 TIOB pin	-	❍
	OUT7		32-bit output compare ch.7 output pin	-	-
	INT15		External interrupt request 15 input pin	-	❍
	SIFD13_1		Slave interface data I/O pin bit13 (Port 1)	❍	-
118	P40	Q*2	General-purpose I/O port	-	❍
	A16		External bus interface address bus bit16	-	-
	SOUT8 (SDA8)		Multifunction serial interface ch.8 output pin. This pin operates as SOUT8 when it is used in a UART/CSIO (operation modes 0 to 2) and as SDA8 when it is used in an I^2C (operation mode 4).	-	❍
	SIFD14_1		Slave interface data I/O pin bit14 (Port 1)	❍	-
119	P41	Q*2	General-purpose I/O port	-	❍
	A17		External bus interface address bus bit17	-	-
	SIN8		Multifunction serial interface ch.8 input pin	-	❍
	SIFD15_1		Slave interface data I/O pin bit15 (Port 1)	❍	-
120	V_{CC}	-	Power pin	-	-

* 1 : Refer to "■ I/O CIRCUIT TYPE" for details on the I/O circuit types.
* 2 : 5 V tolerant pin

2.3 I/O Circuit Types

Table 2.3-1 lists the I/O circuit types for the MB91660 series.

■ I/O circuit types

Table 2.3-1 I/O circuit types (1 / 6)

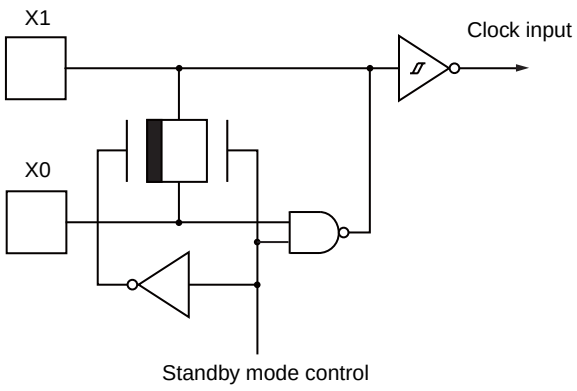
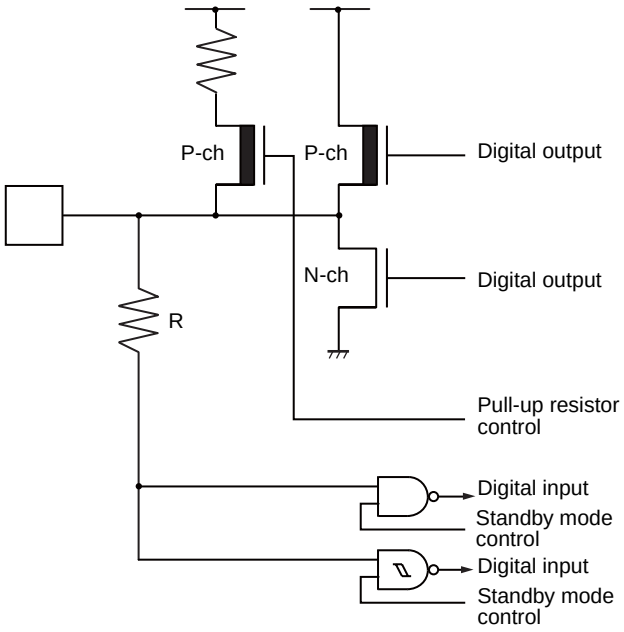
Type	Circuit	Remarks
A		- Oscillation feedback resistor: Approximately 1MΩ - With standby mode control
B		- CMOS level output - CMOS level input - CMOS level hysteresis input - With pull-up resistor control - With standby mode control * CMOS level input when input data, RDY pin and slave interface of external bus interface. Input other than above situations, CMOS level hysteresis input.

Table 2.3-1 I/O circuit types (2 / 6)

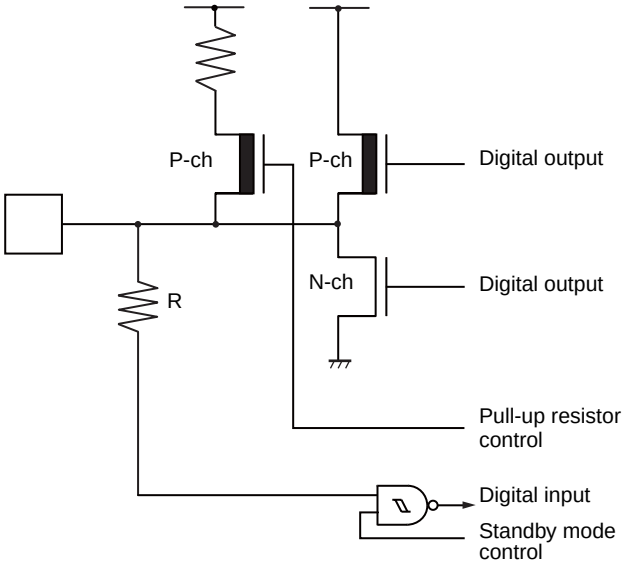
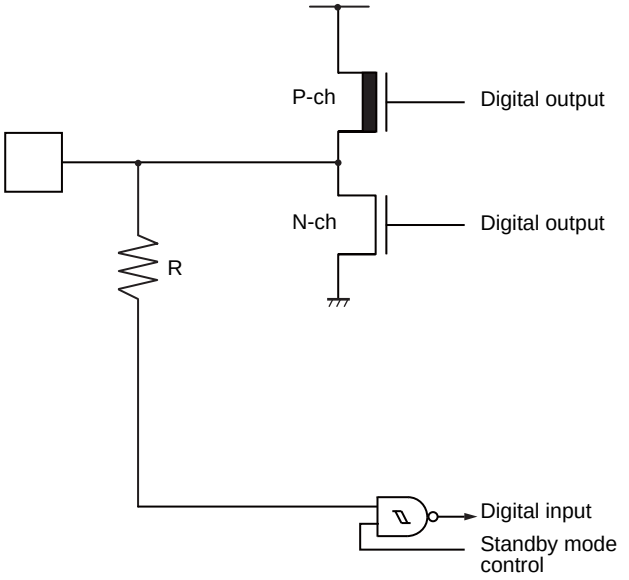
Type	Circuit	Remarks
C		- CMOS level output - CMOS level hysteresis input - With pull-up resistor control - With standby mode control
D		- CMOS level output - CMOS level hysteresis input - 5V tolerant input - With standby mode control

Table 2.3-1 I/O circuit types (3 / 6)

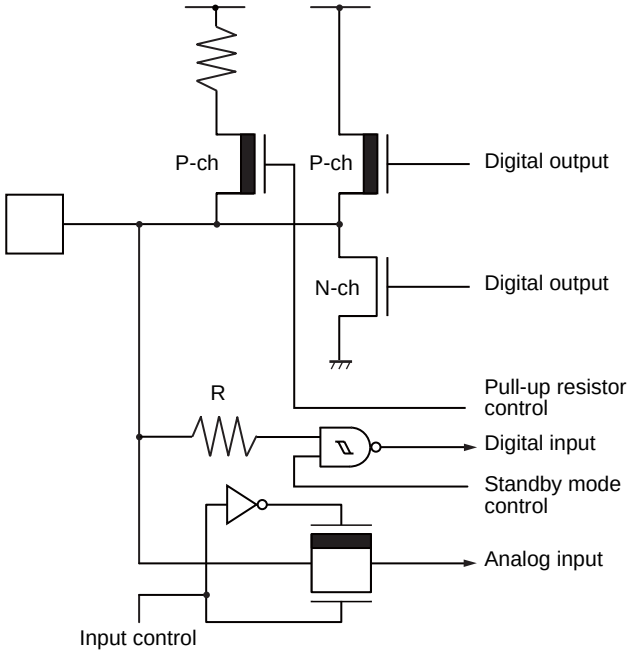
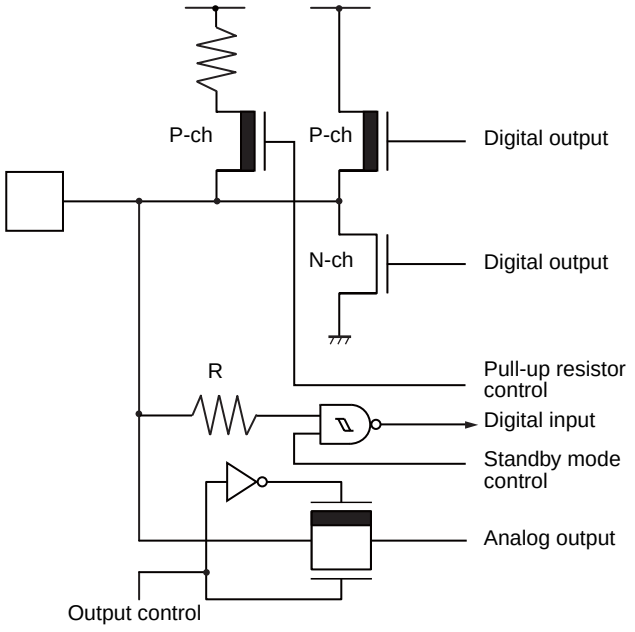
Type	Circuit	Remarks
E		- CMOS level output - CMOS level hysteresis input - With input control - Analog input - With pull-up resistor control - With standby mode control
F		- CMOS level output - CMOS level hysteresis input - With input control - Analog output - With pull-up resistor control - With standby mode control

Table 2.3-1 I/O circuit types (4 / 6)

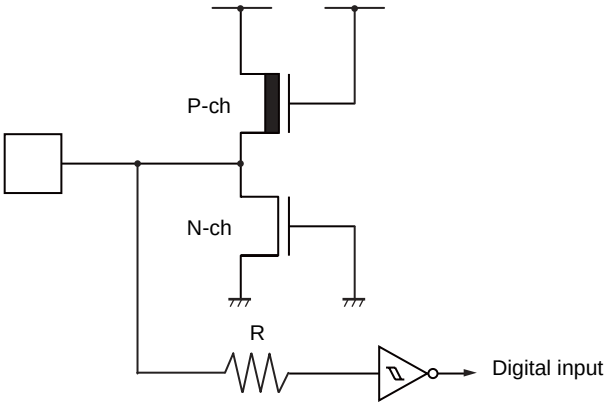
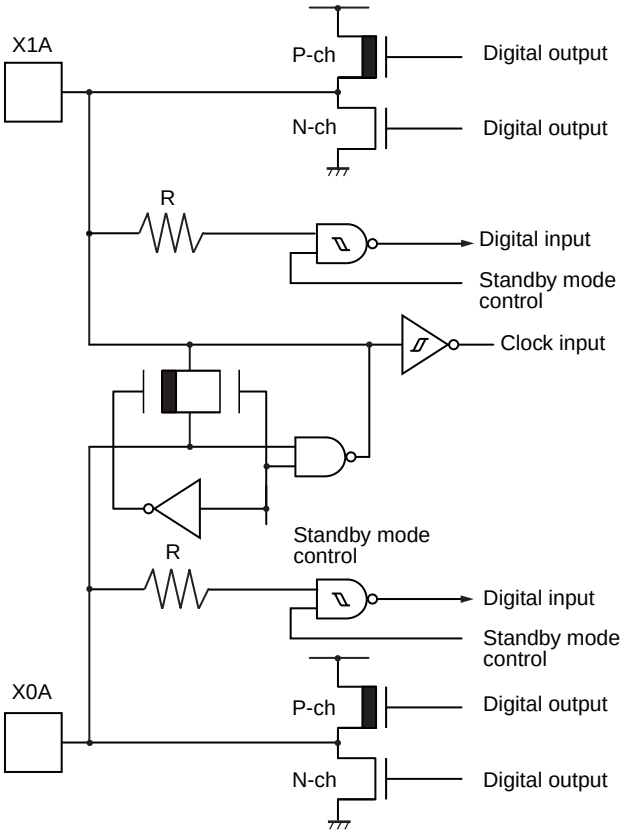
Type	Circuit	Remarks
H		- CMOS level hysteresis input
I		- Oscillation feedback resistor: Approximately 10MΩ - CMOS level output - CMOS level hysteresis input - With standby mode control

Table 2.3-1 I/O circuit types (5 / 6)

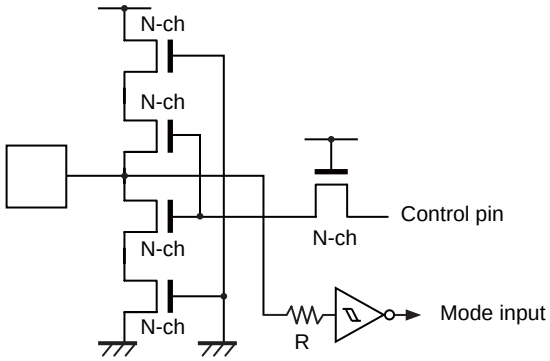
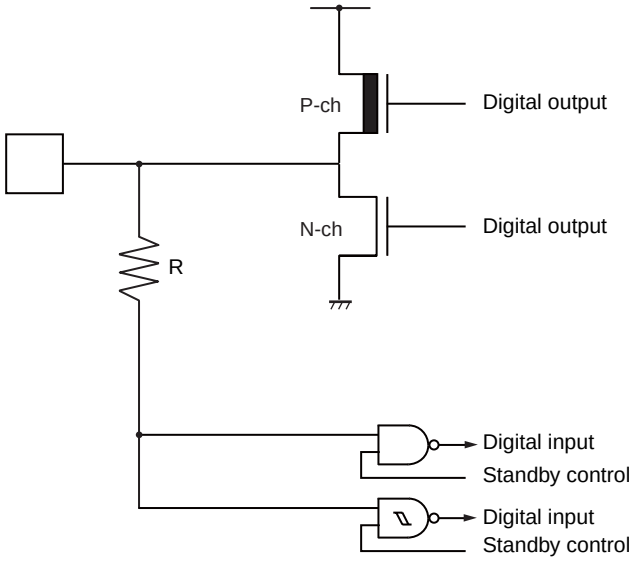
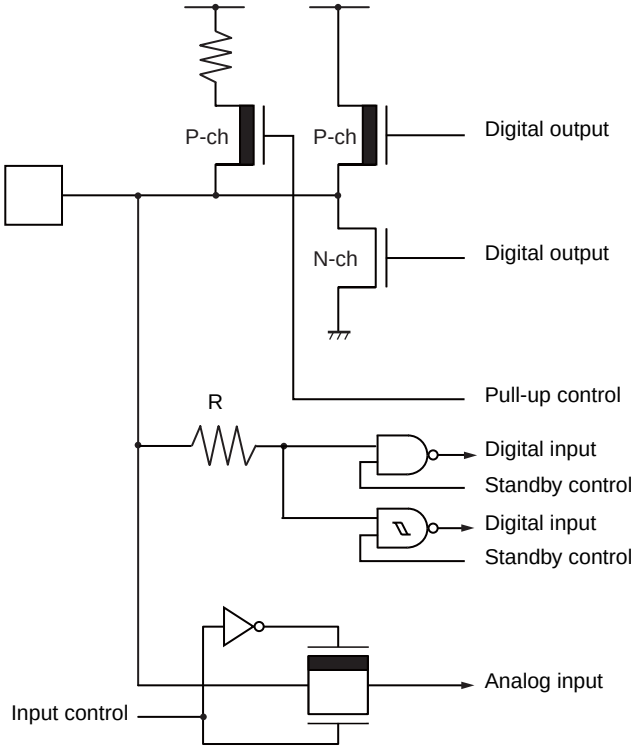
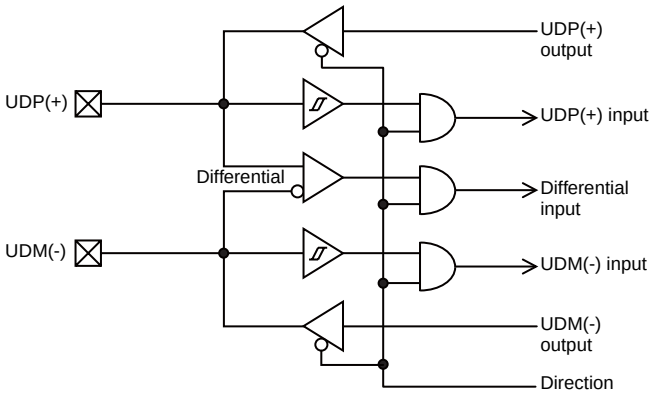
Type	Circuit	Remarks
P		- Flash memory products only - CMOS level hysteresis input - With high-voltage control for flash memory tests
Q		- CMOS level output - CMOS level input - CMOS level hysteresis input - 5 V tolerant input - With standby control

Table 2.3-1 I/O circuit types (6 / 6)

Type	Circuit	Remarks
R		- CMOS level output - CMOS level input - CMOS level hysteresis input - With input control - Analog input - With pull-up control - With standby control
USB		- USB I/O pin

2.4 Setting Method for Pins

This section explains how to set registers for the multiplexed pins.

More than one function has been assigned to the multiplexed pins. The tables below list the register setting values used to assign each of these functions to the pins, as categorized by peripheral function.

The register names appearing in these tables are abbreviated names.

- EPFR: Extended port function register
- PFR: Port function register
- DDR: Port data direction register

For details of these registers, see "CHAPTER 15 I/O Ports".

Other abbreviated register names are explained in the notes under each table. For details, see the respective chapters.

■ Ports

Pin Name	Register Name	Bit Name	Written Value
P00 to P07	PFR0	PFR00 to PFR07	0
P10 to P17	PFR1	PFR10 to PFR17	0
P20 to P27	PFR2	PFR20 to PFR27	0
P30 to P37	PFR3	PFR30 to PFR37	0
P40 to P47	PFR4	PFR40 to PFR47	0
P50 to P57	PFR5	PFR50 to PFR57	0
P60 to P67	PFR6*	PFR60 to PFR67	0
P70 to P77	PFR7	PFR70 to PFR77	0
P80 to P87	PFR8	PFR80 to PFR87	0
PA0 to PA7	PFRA*	PFRA0 to PFRA7	0
PG0 to PG7	PFRG*	PFRG0 to PFRG7	0
PH0 to PH3	PFRH*	PFRH0 to PFRH3	0

*: The PFR register settings for P60, P62, P65, PA5, PG3, PG7, PH1, and PH3 are not required.

<Note>

For details of the settings of the port data direction register (DDR) see "CHAPTER 15 I/O Ports".

■ Clocks

Pin Name	Register Name	Bit Name	Written Value
X0A, X1A	DDRK	DDRK1, DDRK0	00
	EPFR19	XAE	1
	CSELR	SCEN	1

CSELR: Clock source select register

■ External interrupt controllers

One of either of the INTx or INTx_1 pins can be selected for use with each channel.

To use the INT pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the external interrupt controller (For details, see "CHAPTER 16 External Interrupt Controllers").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0 to 3	Port 0	INT0 to INT3	DDR1	DDR10 to DDR13	0
			PFR1	PFR10 to PFR13	0
			EPFR28	INT0E to INT3E	0
	Port 1	INT0_1 to INT3_1	DDRH	DDRH0 to DDRH3	0
			PFRH	*	0
			EPFR28	INT0E to INT3E	1
4 to 7	Port 0	INT4 to INT7	DDR1	DDR14 to DDR17	0
			PFR1	PFR14 to PFR17	0
			EPFR28	INT4E to INT7E	0
8 to 15	Port 0	INT8 to INT15	DDR3	DDR30 to DDR37	0
			PFR3	PFR30 to PFR37	0
			EPFR29	INT8E to INT15E	0

*: INT0_1:PFRH0, INT1_1: no PFR, INT2_1:PFRH2, INT3_1: no PFR

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
16 to 19	Port 0	INT16 to INT19	DDR7	DDR70 to DDR73	0
			PFR7	PFR70 to PFR73	0
			EPFR30	INT16E to INT19E	0
			ADCHE	ADE0 to ADE3	0
	Port 1	INT16_1 to INT19_1	DDRA	DDRA0 to DDRA3	0
			PFRA	PFRA0 to PFRA3	0
			EPFR30	INT16E to INT19E	1
			ADCHE	ADE16 to ADE19	0
20	Port 0	INT20	DDR7	DDR74	0
			PFR7	PFR74	0
			EPFR31	INT20E	0
			ADCHE	ADE4	0
	Port 1	INT20_1	DDRA	DDRA4	0
			PFRA	PFRA4	0
			EPFR31	INT20E	1
			ADCHE	ADE20	0
21	Port 0	INT21	DDR7	DDR75	0
			PFR7	PFR75	0
			EPFR31	INT21E1, INT21E0	00
			ADCHE	ADE5	0
	Port 1	INT21_1	DDRA	DDRA5	0
			EPFR31	INT21E1, INT21E0	01
			ADCHE	ADE21	0
	Port 2	INT21_2	DDR5	DDR53	0
			PFR5	PFR53	0
			EPFR31	INT21E1, INT21E0	10

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
22	Port 0	INT22	DDR7	DDR76	0
			PFR7	PFR76	0
			EPFR31	INT22E1, INT22E0	00
			ADCHE	ADE6	0
	Port 1	INT22_1	DDRA	DDRA6	0
			PFRA	PFRA6	0
			EPFR31	INT22E1, INT22E0	01
			ADCHE	ADE22	0
	Port 2	INT22_2	DDR6	DDR63	0
			PFR6	PFR63	0
			EPFR31	INT22E1, INT22E0	10
23	Port 0	INT23	DDR7	DDR77	0
			PFR7	PFR77	0
			EPFR31	INT23E1, INT23E0	00
			ADCHE	ADE7	0
	Port 1	INT23_1	DDRA	DDRA7	0
			PFRA	PFRA7	0
			EPFR31	INT23E1, INT23E0	01
			ADCHE	ADE23	0
	Port 2	INT23_2	DDR6	DDR67	0
			PFR6	PFR67	0
			EPFR31	INT23E1, INT23E0	10
24 to 31	Port 0	INT24 to INT31	DDR8	DDR80 to DDR87	0
			PFR8	PFR80 to PFR87	0
			EPFR32	INT24E to INT31E	0
			ADCHE	ADE8 to ADE15	0

■ 32-bit free-run timer

The 32-bit free-run timer provides 2 FRCK pins for use with each channel.

One of each of the pins can be selected for use with each channel.

To use the FRCK pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the 32-bit free-run timer (For details, see "CHAPTER 19 32-bit Free-Run Timer").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	FRCK0	DDR5	DDR56	0
			PFR5	PFR56	0
			EPFR34	FRCK0E1, FRCK0E0	00
	Port 1	FRCK0_1	DDR6	DDR66	0
			PFR6	PFR66	0
			EPFR34	FRCK0E1, FRCK0E0	01
1	Port 0	FRCK1	DDR5	DDR53	0
			PFR5	PFR53	0
			EPFR34	FRCK1E1, FRCK1E0	00
	Port 1	FRCK1_1	DDR6	DDR63	0
			PFR6	PFR63	0
			EPFR34	FRCK1E1, FRCK1E0	01

■ 32-bit input capture

The 32-bit input capture provides 3 IN pins for use with each channel.

One of each of the pins can be selected for use with each channel.

To use the IN pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the 32-bit input capture (For details, see "CHAPTER 20 32-bit Input Capture").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	IN0	DDR0	DDR00	0
			PFR0	PFR00	0
			EPFR4	IN0E1, IN0E0	00
	Port 1	IN0_1	DDR8	DDR80	0
			PFR8	PFR80	0
			EPFR4	IN0E1, IN0E0	01
			ADCHE	ADE8	0
	Port 2	IN0_2	DDRG	DDRG0	0
			PFRG	PFRG0	0
			EPFR4	IN0E1, IN0E0	10
1	Port 0	IN1	DDR0	DDR01	0
			PFR0	PFR01	0
			EPFR4	IN1E1, IN1E0	00
	Port 1	IN1_1	DDR8	DDR81	0
			PFR8	PFR81	0
			EPFR4	IN1E1, IN1E0	01
			ADCHE	ADE9	0
	Port 2	IN1_2	DDRG	DDRG1	0
			PFRG	PFRG1	0
			EPFR4	IN1E1, IN1E0	10

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
2	Port 0	IN2	DDR0	DDR02	0
			PFR0	PFR02	0
			EPFR4	IN2E1, IN2E0	00
	Port 1	IN2_1	DDR8	DDR82	0
			PFR8	PFR82	0
			EPFR4	IN2E1, IN2E0	01
			ADCHE	ADE10	0
	Port 2	IN2_2	DDRG	DDRG2	0
			PFRG	PFRG2	0
			EPFR4	IN2E1, IN2E0	10
3	Port 0	IN3	DDR0	DDR03	0
			PFR0	PFR03	0
			EPFR4	IN3E1, IN3E0	00
	Port 1	IN3_1	DDR8	DDR83	0
			PFR8	PFR83	0
			EPFR4	IN3E1, IN3E0	01
			ADCHE	ADE11	0
	Port 2	IN3_2	DDRG	DDRG3	0
			EPFR4	IN3E1, IN3E0	10
4	Port 0	IN4	DDR0	DDR04	0
			PFR0	PFR04	0
			EPFR5	IN4E1, IN4E0	00
	Port 1	IN4_1	DDR8	DDR84	0
			PFR8	PFR84	0
			EPFR5	IN4E1, IN4E0	01
			ADCHE	ADE12	0
	Port 2	IN4_2	DDRG	DDRG4	0
			PFRG	PFRG4	0
			EPFR5	IN4E1, IN4E0	10

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
5	Port 0	IN5	DDR0	DDR05	0
			PFR0	PFR05	0
			EPFR5	IN5E1, IN5E0	00
	Port 1	IN5_1	DDR8	DDR85	0
			PFR8	PFR85	0
			EPFR5	IN5E1, IN5E0	01
			ADCHE	ADE13	0
	Port 2	IN5_2	DDRG	DDRG5	0
			PFRG	PFRG5	0
			EPFR5	IN5E1, IN5E0	10
6	Port 0	IN6	DDR0	DDR06	0
			PFR0	PFR06	0
			EPFR5	IN6E1, IN6E0	00
	Port 1	IN6_1	DDR8	DDR86	0
			PFR8	PFR86	0
			EPFR5	IN6E1, IN6E0	01
			ADCHE	ADE14	0
	Port 2	IN6_2	DDRG	DDRG6	0
			PFRG	PFRG6	0
			EPFR5	IN6E1, IN6E0	10
7	Port 0	IN7	DDR0	DDR07	0
			PFR0	PFR07	0
			EPFR5	IN7E1, IN7E0	00
	Port 1	IN7_1	DDR8	DDR87	0
			PFR8	PFR87	0
			EPFR5	IN7E1, IN7E0	01
			ADCHE	ADE15	0
	Port 2	IN7_2	DDRG	DDRG7	0
			EPFR5	IN7E1, IN7E0	10

■ 32-bit output compare

The 32-bit output compare provides 2 OUT pins for use with each channel.

One of each of the pins can be selected for use with each channel.

To use the OUT pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	OUT0	PFR2	PFR24	1
			EPFR0	OUT0E2 to OUT0E0	001
			EPFR25	TIOA10E1, TIOA10E0	Other than 01*
			EPFR11	SOUT5E1, SOUT5E0	Other than 01*
	Port 1	OUT0_1	PFR7	PFR70	1
			EPFR0	OUT0E2 to OUT0E0	010
			ADCHE	ADE0	0
1	Port 0	OUT1	PFR2	PFR25	1
			EPFR0	OUT1E2 to OUT1E0	001
	Port 1	OUT1_1	PFR7	PFR71	1
			EPFR0	OUT1E2 to OUT1E0	010
			ADCHE	ADE1	0
2	Port 0	OUT2	PFR2	PFR26	1
			EPFR1	OUT2E2 to OUT2E0	001
			EPFR25	TIOA11E1, TIOA11E0	Other than 01*
			EPFR11	SCK5E1, SCK5E0	Other than 01*
	Port 1	OUT2_1	PFR7	PFR72	1
			EPFR1	OUT2E2 to OUT2E0	010
			EPFR33	TMO0E1, TMO0E0	Other than 01*
			ADCHE	ADE2	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
3	Port 0	OUT3	PFR2	PFR27	1
			EPFR1	OUT3E2 to OUT3E0	001
	Port 1	OUT3_1	PFR7	PFR73	1
			EPFR1	OUT3E2 to OUT3E0	010
			EPFR33	TMO1E1, TMO1E0	Other than 01*
			ADCHE	ADE3	0
4	Port 0	OUT4	PFR3	PFR34	1
			EPFR2	OUT4E2 to OUT4E0	001
			EPFR27	TIOA14E1,TIOA14E0	Other than 01*
			EPFR13	SOUT7E1, SOUT7E0	Other than 01*
	Port 1	OUT4_1	PFR7	PFR74	1
			EPFR2	OUT4E2 to OUT4E0	010
			EPFR34	TMO2E1, TMO2E0	Other than 01*
			ADCHE	ADE4	0
5	Port 0	OUT5	PFR3	PFR35	1
			EPFR2	OUT5E2 to OUT5E0	001
	Port 1	OUT5_1	PFR7	PFR75	1
			EPFR2	OUT5E2 to OUT5E0	010
			EPFR6	SOUT0E2 to SOUT0E0	Other than 001*
			ADCHE	ADE5	0
6	Port 0	OUT6	PFR3	PFR36	1
			EPFR3	OUT6E2 to OUT6E0	001
			EPFR27	TIOA15E1, TIOA15E0	Other than 01*
			EPFR13	SCK7E1, SCK7E0	Other than 01*
	Port 1	OUT6_1	PFR7	PFR76	1
			EPFR3	OUT6E2 to OUT6E0	010
			ADCHE	ADE6	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
7	Port 0	OUT7	PFR3	PFR37	1
			EPFR3	OUT7E2 to OUT7E0	001
	Port 1	OUT7_1	PFR7	PFR77	1
			EPFR3	OUT7E2 to OUT7E0	010
			EPFR6	SCK0E2 to SCK0E0	Other than 001*
			ADCHE	ADE7	0

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

■ 16-bit reload timer

The 16-bit reload timer provides 2 of each of the TMI/TMO pins for use with each channel.

One of each of the TMI/TMO pins can be selected for use with each channel. However, to use pins for the same channel, the pins must be assigned to the same port number.

To use the TMI pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the 16-bit reload timer (For details, see "CHAPTER 22 16-bit Reload Timer").

To use the TMO pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	TMI0	DDR7	DDR75	0
			PFR7	PFR75	0
			EPFR33	TMI0E	0
			ADCHE	ADE5	0
		TMO0	PFR7	PFR72	1
			EPFR33	TMO0E1, TMO0E0	01
			ADCHE	ADE2	0
	Port 1	TMI0_1	DDRA	DDRA5	0
			EPFR33	TMI0E	1
			ADCHE	ADE21	0
		TMO0_1	PFRA	PFRA2	1
			EPFR33	TMO0E1, TMO0E0	10
			ADCHE	ADE18	0
1	Port 0	TMI1	DDR7	DDR76	0
			PFR7	PFR76	0
			EPFR33	TMI1E	0
			ADCHE	ADE6	0
		TMO1	PFR7	PFR73	1
			EPFR33	TMO1E1, TMO1E0	01
			ADCHE	ADE3	0
	Port 1	TMI1_1	DDRA	DDRA6	0
			PFRA	PFRA6	0
			EPFR33	TMI1E	1
			ADCHE	ADE22	0
		TMO1_1	PFRA	PFRA3	1
			EPFR33	TMO1E1, TMO1E0	10
			ADCHE	ADE19	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
2	Port 0	TMI2	DDR7	DDR77	0
			PFR7	PFR77	0
			EPFR34	TMI2E	0
			ADCHE	ADE7	0
		TMO2	PFR7	PFR74	1
			EPFR34	TMO2E1, TMO2E0	01
			ADCHE	ADE4	0
	Port 1	TMI2_1	DDRA	DDRA7	0
			PFRA	PFRA7	0
			EPFR34	TMI2E	1
			ADCHE	ADE23	0
		TMO2_1	PFRA	PFRA4	1
			EPFR34	TMO2E1, TMO2E0	10
			ADCHE	ADE20	0

■ Base timer

The base timer provides 2 TIOA/TIOB pins for use with channels 0 to 5 and 1 TIOA/TIOB pin for use with channels 6 to 15.

One of each of the TIOA/TIOB pins can be selected for use with channels 0 to 5. However, to use pins for the same channel, the pins must be assigned to the same port number.

To use the TIOA/TIOB pins for input, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the base timer (For details, see "CHAPTER 24 Base Timer").

To use the TIOA pin for output, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	TIOA0	PFR0	PFR00	1
			EPFR20	TIOA0E1, TIOA0E0	01
			EPFR6	SOUT0E2 to SOUT0E0	Other than 010 *
		TIOB0	PFR0	PFR01	0
			DDR0	DDR01	0
			EPFR20	TIOB0E	0
	Port 1	TIOA0_1	PFRG	PFRG0	1
			EPFR20	TIOA0E1, TIOA0E0	10
			EPFR6	SOUT0E2 to SOUT0E0	Other than 100 *
		TIOB0_1	PFRG	PFRG1	0
			DDRG	DDRG1	0
			EPFR20	TIOB0E	1
1	Port 0	TIOA1	PFR0	PFR02	At input: 0 At output: 1
			DDR0	DDR02	0 (only at input)
			EPFR20	TIOA1E1, TIOA1E0	01
			EPFR6	SCK0E2 to SCK0E0	Other than 010 *
		TIOB1	PFR0	PFR03	0
			DDR0	DDR03	0
			EPFR20	TIOB1E	0
	Port 1	TIOA1_1	PFRG	PFRG2	At input: 0 At output: 1
			DDRG	DDRG2	0 (only at input)
			EPFR20	TIOA1E1, TIOA1E0	10
			EPFR6	SCK0E2 to SCK0E0	Other than 100 *
		TIOB1_1	DDRG	DDRG3	0
			EPFR20	TIOB1E	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
2	Port 0	TIOA2	PFR0	PFR04	1
			EPFR21	TIOA2E1, TIOA2E0	01
			EPFR7	SOUT1E1, SOUT1E0	Other than 01 *
		TIOB2	PFR0	PFR05	0
			DDR0	DDR05	0
			EPFR21	TIOB2E	0
	Port 1	TIOA2_1	PFRG	PFRG4	1
			EPFR21	TIOA2E1, TIOA2E0	10
			EPFR7	SOUT1E1, SOUT1E0	Other than 10 *
		TIOB2_1	PFRG	PFRG5	0
			DDRG	DDRG5	0
			EPFR21	TIOB2E	1
3	Port 0	TIOA3	PFR0	PFR06	At input: 0 At output: 1
			DDR0	DDR06	0 (only at input)
			EPFR21	TIOA3E1, TIOA3E0	01
			EPFR7	SCK1E1, SCK1E0	Other than 01 *
		TIOB3	PFR0	PFR07	0
			DDR0	DDR07	0
			EPFR21	TIOB3E	0
	Port 1	TIOA3_1	PFRG	PFRG6	At input: 0 At output: 1
			DDRG	DDRG6	0 (only at input)
			EPFR21	TIOA3E1, TIOA3E0	10
			EPFR7	SCK1E1, SCK1E0	Other than 10 *
		TIOB3_1	DDRG	DDRG7	0
			EPFR21	TIOB3E	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
4	Port 0	TIOA4	PFR1	PFR10	1
			EPFR22	TIOA4E1, TIOA4E0	01
			EPFR8	SOUT2E1, SOUT2E0	Other than 01 *
		TIOB4	PFR1	PFR11	0
			DDR1	DDR11	0
			EPFR22	TIOB4E	0
	Port 1	TIOA4_1	PFRH	PFRH0	1
			EPFR22	TIOA4E1, TIOA4E0	10
			EPFR8	SOUT2E1, SOUT2E0	Other than 10 *
		TIOB4_1	DDRH	DDRH1	0
			EPFR22	TIOB4E	1
5	Port 0	TIOA5	PFR1	PFR12	At input: 0 At output: 1
			DDR1	DDR12	0 (only at input)
			EPFR22	TIOA5E1, TIOA5E0	01
			EPFR8	SCK2E1, SCK2E0	Other than 01 *
		TIOB5	PFR1	PFR13	0
			DDR1	DDR13	0
			EPFR22	TIOB5E	0
	Port 1	TIOA5_1	PFRH	PFRH2	At input: 0 At output: 1
			DDRH	DDRH2	0 (only at input)
			EPFR22	TIOA5E1, TIOA5E0	10
			EPFR8	SCK2E1, SCK2E0	Other than 10 *
		TIOB5_1	DDRH	DDRH3	0
			EPFR22	TIOB5E	1
6	Port 0	TIOA6	PFR1	PFR14	1
			EPFR23	TIOA6E1, TIOA6E0	01
			EPFR9	SOUT3E1, SOUT3E0	Other than 01*
		TIOB6	PFR1	PFR15	0
			DDR1	DDR15	0
			EPFR23	TIOB6E	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
7	Port 0	TIOA7	PFR1	PFR16	At input: 0 At output: 1
			DDR1	DDR16	0 (only at input)
			EPFR23	TIOA7E1, TIOA7E0	01
			EPFR9	SCK3E1, SCK3E0	Other than 01 *
		TIOB7	PFR1	PFR17	0
			DDR1	DDR17	0
			EPFR23	TIOB7E	0
8	Port 0	TIOA8	PFR2	PFR20	1
			EPFR24	TIOA8E1, TIOA8E0	01
			EPFR10	SOUT4E1, SOUT4E0	Other than 01 *
		TIOB8	PFR2	PFR21	0
			DDR2	DDR21	0
			EPFR24	TIOB8E	0
9	Port 0	TIOA9	PFR2	PFR22	At input: 0 At output: 1
			DDR2	DDR22	0 (only at input)
			EPFR24	TIOA9E1, TIOA9E0	01
			EPFR10	SCK4E1, SCK4E0	Other than 01 *
		TIOB9	PFR2	PFR23	0
			DDR2	DDR23	0
			EPFR24	TIOB9E	0
10	Port 0	TIOA10	PFR2	PFR24	1
			EPFR25	TIOA10E1, TIOA10E0	01
			EPFR11	SOUT5E1, SOUT5E0	Other than 01 *
		TIOB10	PFR2	PFR25	0
			DDR2	DDR25	0
			EPFR25	TIOB10E	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
11	Port 0	TIOA11	PFR2	PFR26	At input: 0 At output: 1
			DDR2	DDR26	0 (only at input)
			EPFR25	TIOA11E1, TIOA11E0	01
			EPFR11	SCK5E1, SCK5E0	Other than 01 *
		TIOB11	PFR2	PFR27	0
			DDR2	DDR27	0
			EPFR25	TIOB11E	0
12	Port 0	TIOA12	PFR3	PFR30	1
			EPFR26	TIOA12E1, TIOA12E0	01
			EPFR12	SOUT6E1, SOUT6E0	Other than 01 *
		TIOB12	PFR3	PFR31	0
			DDR3	DDR31	0
			EPFR26	TIOB12E	0
13	Port 0	TIOA13	PFR3	PFR32	At input: 0 At output: 1
			DDR3	DDR32	0 (only at input)
			EPFR26	TIOA13E1, TIOA13E0	01
			EPFR12	SCK6E1, SCK6E0	Other than 01 *
		TIOB13	PFR3	PFR33	0
			DDR3	DDR33	0
			EPFR26	TIOB13E	0
14	Port 0	TIOA14	PFR3	PFR34	1
			EPFR27	TIOA14E1, TIOA14E0	01
			EPFR13	SOUT7E1, SOUT7E0	Other than 01 *
		TIOB14	PFR3	PFR35	0
			DDR3	DDR35	0
			EPFR27	TIOB14E	0

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
15	Port 0	TIOA15	PFR3	PFR36	At input: 0 At output: 1
			DDR3	DDR36	0 (only at input)
			EPFR27	TIOA15E1, TIOA15E0	01
			EPFR13	SCK7E1, SCK7E0	Other than 01 *
		TIOB15	PFR3	PFR37	0
			DDR3	DDR37	0
			EPFR27	TIOB15E	0

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

■ Up/Down counter

The up/down counter provides 3 AIN/BIN/ZIN pins for use with channel 0, and it provides 2 AIN/BIN/ZIN pins for use with channels 1 to 3.

One of each of the AIN/BIN/ZIN pins can be selected for use with each channel. However, to use pins for the same channel, the pins must be assigned to the same port number.

To use the AIN/BIN/ZIN pins, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the up/down counter (For details, see "CHAPTER 25 Up/Down Counter").

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	AIN0	DDR1	DDR10	0
			PFR1	PFR10	0
			EPFR18	UDIN0E1, UDIN0E0	00
		BIN0	DDR1	DDR11	0
			PFR1	PFR11	0
			EPFR18	UDIN0E1, UDIN0E0	00
		ZIN0	DDR1	DDR12	0
			PFR1	PFR12	0
			EPFR18	UDIN0E1, UDIN0E0	00
	Port 1	AIN0_1	DDR5	DDR50	0
			PFR5	PFR50	0
			EPFR18	UDIN0E1, UDIN0E0	01
		BIN0_1	DDR5	DDR51	0
			PFR5	PFR51	0
			EPFR18	UDIN0E1, UDIN0E0	01
		ZIN0_1	DDR5	DDR52	0
			PFR5	PFR52	0
			EPFR18	UDIN0E1, UDIN0E0	01
	Port 2	AIN0_2	DDRH	DDRH0	0
			PFRH	PFRH0	0
			EPFR18	UDIN0E1, UDIN0E0	10
		BIN0_2	DDRH	DDRH1	0
			EPFR18	UDIN0E1, UDIN0E0	10
		ZIN0_2	DDRH	DDRH2	0
			PFRH	PFRH2	0
			EPFR18	UDIN0E1, UDIN0E0	10

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
1	Port 0	AIN1	DDR1	DDR14	0
			PFR1	PFR14	0
			EPFR18	UDIN1E1, UDIN1E0	00
		BIN1	DDR1	DDR15	0
			PFR1	PFR15	0
			EPFR18	UDIN1E1, UDIN1E0	00
		ZIN1	DDR1	DDR16	0
			PFR1	PFR16	0
			EPFR18	UDIN1E1, UDIN1E0	00
	Port 1	AIN1_1	DDR5	DDR54	0
			PFR5	PFR54	0
			EPFR18	UDIN1E1, UDIN1E0	01
		BIN1_1	DDR5	DDR55	0
			PFR5	PFR55	0
			EPFR18	UDIN1E1, UDIN1E0	01
		ZIN1_1	DDR5	DDR56	0
			PFR5	PFR56	0
			EPFR18	UDIN1E1, UDIN1E0	01

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
2	Port 0	AIN2	DDR2	DDR20	0
			PFR2	PFR20	0
			EPFR18	UDIN2E1, UDIN2E0	00
		BIN2	DDR2	DDR21	0
			PFR2	PFR21	0
			EPFR18	UDIN2E1, UDIN2E0	00
		ZIN2	DDR2	DDR22	0
			PFR2	PFR22	0
			EPFR18	UDIN2E1, UDIN2E0	00
	Port 1	AIN2_1	DDR6	DDR60	0
			EPFR18	UDIN2E1, UDIN2E0	01
		BIN2_1	DDR6	DDR61	0
			PFR6	PFR61	0
			EPFR18	UDIN2E1, UDIN2E0	01
		ZIN2_1	DDR6	DDR62	0
			EPFR18	UDIN2E1, UDIN2E0	01

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
3	Port 0	AIN3	DDR2	DDR24	0
			PFR2	PFR24	0
			EPFR18	UDIN3E1, UDIN3E0	00
		BIN3	DDR2	DDR25	0
			PFR2	PFR25	0
			EPFR18	UDIN3E1, UDIN3E0	00
		ZIN3	DDR2	DDR26	0
			PFR2	PFR26	0
			EPFR18	UDIN3E1, UDIN3E0	00
	Port 1	AIN3_1	DDR6	DDR64	0
			PFR6	PFR64	0
			EPFR18	UDIN3E1, UDIN3E0	01
		BIN3_1	DDR6	DDR65	0
			EPFR18	UDIN3E1, UDIN3E0	01
		ZIN3_1	DDR6	DDR66	0
			PFR6	PFR66	0
			EPFR18	UDIN3E1, UDIN3E0	01

■ 10-bit A/D converter

- AN pins

Pin Name	Register Name	Bit Name	Written Value
AN0 to AN7	ADCHE	ADE0 to ADE7	1
AN8 to AN15	ADCHE	ADE8 to ADE15	1
AN16 to AN23	ADCHE	ADE16 to ADE23	1

ADCHE: A/D channel enable register

- ADTRG0 pins

The 10-bit A/D converter provides 4 ADTRG0 pins.

One of each of the pins can be selected for use with each channel.

To use the ADTRG0 pin, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the 10-bit A/D converter (For details, see "CHAPTER 26 10-Bit A/D Converter").

For details of the basic settings, see the following table.

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	ADTRG0	DDR5	DDR55	0
		PFR5	PFR55	0
		EPFR19	ADTRG0E2 to ADTRG0E0	000
Port 1	ADTRG0_1	DDR6	DDR65	0
		EPFR19	ADTRG0E2 to ADTRG0E0	001
Port 2	ADTRG0_2	DDRK	DDRK2	0
		EPFR19	ADTRG0E2 to ADTRG0E0	010
Port 3	ADTRG0_3	DDRK	DDRK3	0
		EPFR19	ADTRG0E2 to ADTRG0E0	011

■ 8-bit D/A converter

Pin Name	Register Name	Bit Name	Written Value
DA0 to DA2	DACR0 to DACR2	DAE	1

DACR: D/A control register

■ Multifunction serial interface

The multifunction serial interface provides multiple SCK pins , SIN pins, and SOUT pins for use with one channel.

One of each of the SCK/SIN/SOUT pins can be selected for use with each channel. However, to use pins for the same channel, the pins must be assigned to the same port number.

To use the SIN/SCK pins for input, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the multifunction serial interface (For details, see "CHAPTER 28 Multi-function Serial Interface").

To use the SOUT/SCK pins for output, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 0	SCK0	PFR7	PFR77	At SCK input: 0 At SCK output: 1
			DDR7	DDR77	0 (only at SCK input)
			EPFR6	SCK0E2 to SCK0E0	001
			SMR0	SCKE	Input enable: 0 Output enable: 1
			ADCHE	ADE7	0
		SIN0	DDR7	DDR76	0
			PFR7	PFR76	0
			EPFR6	SIN0E1, SIN0E0	00
			ADCHE	ADE6	0
		SOUT0	PFR7	PFR75	1
			EPFR6	SOUT0E2 to SOUT0E0	001
			SMR0	SOE	1
			ADCHE	ADE5	0
	Port 1	SCK0_1	PFR0	PFR02	At SCK input: 0 At SCK output: 1
			DDR0	DDR02	0 (only at SCK input)
			EPFR6	SCK0E2 to SCK0E0	010
			SMR0	SCKE	Input enable: 0 Output enable: 1
		SIN0_1	DDR0	DDR01	0
			PFR0	PFR01	0
			EPFR6	SIN0E1, SIN0E0	01
		SOUT0_1	PFR0	PFR00	1
			EPFR6	SOUT0E2 to SOUT0E0	010
			SMR0	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
0	Port 2	SCK0_2	PFRG	PFRG2	At SCK input: 0 At SCK output: 1
			DDRG	DDRG2	0 (only at SCK input)
			EPFR6	SCK0E2 to SCK0E0	100
			SMR0	SCKE	Input enable: 0 Output enable: 1
		SIN0_2	DDRG	DDRG1	0
			PFRG	PFRG1	0
			EPFR6	SIN0E1, SIN0E0	10
		SOUT0_2	PFRG	PFRG0	1
			EPFR6	SOUT0E2 to SOUT0E0	100
			SMR0	SOE	1
1	Port 0	SCK1 (SCL1)	PFR0	PFR06	At SCK input:0 At SCK output or SCL:1
			DDR0	DDR06	0 (only at SCK input)
			EPFR7	SCK1E1, SCK1E0	01
			SMR1	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN1	DDR0	DDR05	0
			PFR0	PFR05	0
			EPFR7	SIN1E	0
		SOUT1 (SDA1)	PFR0	PFR04	1
			EPFR7	SOUT1E1, SOUT1E0	01
			SMR1	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
1	Port 1	SCK1_1 (SCL1_1)	PFRG	PFRG6	At SCK input:0 At SCK output or SCL:1
			DDRG	DDRG6	0 (only at SCK input)
			EPFR7	SCK1E1, SCK1E0	10
			SMR1	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN1_1	DDRG	DDRG5	0
			PFRG	PFRG5	0
			EPFR7	SIN1E	1
		SOUT1_1 (SDA1_1)	PFRG	PFRG4	1
			EPFR7	SOUT1E1, SOUT1E0	10
			SMR1	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
2	Port 0	SCK2 (SCL2)	PFR1	PFR12	At SCK input:0 At SCK output or SCL:1
			DDR1	DDR12	0 (only at SCK input)
			EPFR8	SCK2E1, SCK2E0	01
			SMR2	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN2	DDR1	DDR11	0
			PFR1	PFR11	0
			EPFR8	SIN2E	0
		SOUT2 (SDA2)	PFR1	PFR10	1
			EPFR8	SOUT2E1, SOUT2E0	01
			SMR2	SOE	1
	Port 1	SCK2_1 (SCL2_1)	PFRH	PFRH2	At SCK input:0 At SCK output or SCL:1
			DDRH	DDRH2	0 (only at SCK input)
			EPFR8	SCK2E1, SCK2E0	10
			SMR2	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN2_1	DDRH	DDRH1	0
			EPFR8	SIN2E	1
		SOUT2_1 (SDA2_1)	PFRH	PFRH0	1
			EPFR8	SOUT2E1, SOUT2E0	10
			SMR2	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
3	Port 0	SCK3 (SCL3)	PFR1	PFR16	At SCK input:0 At SCK output or SCL:1
			DDR1	DDR16	0 (only at SCK input)
			EPFR9	SCK3E1, SCK3E0	01
			SMR3	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN3	DDR1	DDR15	0
			PFR1	PFR15	0
			EPFR9	SIN3E	0
		SOUT3 (SDA3)	PFR1	PFR14	1
			EPFR9	SOUT3E1, SOUT3E0	01
			SMR3	SOE	1
4	Port 0	SCK4 (SCL4)	PFR2	PFR22	At SCK input:0 At SCK output or SCL:1
			DDR2	DDR22	0 (only at SCK input)
			EPFR10	SCK4E1, SCK4E0	01
			SMR4	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN4	DDR2	DDR21	0
			PFR2	PFR21	0
			EPFR10	SIN4E	0
		SOUT4 (SDA4)	PFR2	PFR20	1
			EPFR10	SOUT4E1, SOUT4E0	01
			SMR4	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
5	Port 0	SCK5 (SCL5)	PFR2	PFR26	At SCK input:0 At SCK output or SCL:1
			DDR2	DDR26	0 (only at SCK input)
			EPFR11	SCK5E1, SCK5E0	01
			SMR5	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN5	DDR2	DDR25	0
			PFR2	PFR25	0
			EPFR11	SIN5E	0
		SOUT5 (SDA5)	PFR2	PFR24	1
			EPFR11	SOUT5E1, SOUT5E0	01
			SMR5	SOE	1
6	Port 0	SCK6 (SCL6)	PFR3	PFR32	At SCK input:0 At SCK output or SCL:1
			DDR3	DDR32	0 (only at SCK input)
			EPFR12	SCK6E1, SCK6E0	01
			SMR6	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN6	DDR3	DDR31	0
			PFR3	PFR31	0
			EPFR12	SIN6E	0
		SOUT6 (SDA6)	PFR3	PFR30	1
			EPFR12	SOUT6E1, SOUT6E0	01
			SMR6	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
7	Port 0	SCK7 (SCL7)	PFR3	PFR36	At SCK input:0 At SCK output or SCL:1
			DDR3	DDR36	0 (only at SCK input)
			EPFR13	SCK7E1, SCK7E0	01
			SMR7	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN7	DDR3	DDR35	0
			PFR3	PFR35	0
			EPFR13	SIN7E	0
		SOUT7 (SDA7)	PFR3	PFR34	1
			EPFR13	SOUT7E1, SOUT7E0	01
			SMR7	SOE	1
8	Port 0	SCK8 (SCL8)	PFR4	PFR42	At SCK input:0 At SCK output or SCL:1
			DDR4	DDR42	0 (only at SCK input)
			EPFR14	SCK8E1, SCK8E0	01
			SMR8	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN8	DDR4	DDR41	0
			PFR4	PFR41	0
			EPFR14	SIN8E	0
		SOUT8 (SDA8)	PFR4	PFR40	1
			EPFR14	SOUT8E1, SOUT8E0	01
			SMR8	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
9	Port 0	SCK9 (SCL9)	PFR4	PFR46	At SCK input:0 At SCK output or SCL:1
			DDR4	DDR46	0 (only at SCK input)
			EPFR15	SCK9E1, SCK9E0	01
			SMR9	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN9	DDR4	DDR45	0
			PFR4	PFR45	0
			EPFR15	SIN9E	0
		SOUT9 (SDA9)	PFR4	PFR44	1
			EPFR15	SOUT9E1, SOUT9E0	01
			SMR9	SOE	1
10	Port 0	SCK10 (SCL10)	PFR5	PFR52	At SCK input:0 At SCK output or SCL:1
			DDR5	DDR52	0 (only at SCK input)
			EPFR16	SCK10E1, SCK10E0	01
			SMR10	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN10	DDR5	DDR51	0
			PFR5	PFR51	0
			EPFR16	SIN10E	0
		SOUT10 (SDA10)	PFR5	PFR50	1
			EPFR16	SOUT10E1, SOUT10E0	01
			SMR10	SOE	1

Channel	Port Number	Pin Name	Register Name	Bit Name	Written Value
11	Port 0	SCK11 (SCL11)	PFR5	PFR56	At SCK input:0 At SCK output or SCL:1
			DDR5	DDR56	0 (only at SCK input)
			EPFR17	SCK11E1, SCK11E0	01
			SMR11	SCKE	Input enable:0 Output enable:1 (only at SCK)
		SIN11	DDR5	DDR55	0
			PFR5	PFR55	0
			EPFR17	SIN11E	0
		SOUT11 (SDA11)	PFR5	PFR54	1
			EPFR17	SOUT11E1, SOUT11E0	01
			SMR11	SOE	1

SMR: Serial mode register

<Note>

Different pins are enabled depending on the operation mode. For details, see "CHAPTER 28 Multi-function Serial Interface".

■ DMA controller (DMAC)

To use the DREQ pin for input, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Enable the operation of the DREQ pin (For details, see "CHAPTER 33 DMA Controller (DMAC)").

To use the DACK/DEOP pins for output, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register.
   (For details of the multiplexed pins, see the pin assignment diagram.)
3. Set peripheral functions on the PFR register (PFR=1).

For details of the basic settings, see the following table.

Channel	Pin Name	Register Name	Bit Name	Written Value
0	DACK0	PFR6	PFR63	1
	DEOP0	PFR6	PFR64	1
	DREQ0	DDR6	DDR62	0
1	DACK1	PFR6	PFR66	1
	DEOP1	PFR6	PFR67	1
	DREQ1	DDR6	DDR65	0
2	DACK2	PFRG	PFRG1	1
	DEOP2	PFRG	PFRG2	1
		EPFR20	TIOA1E1, TIOA1E0	Other than 10 *
		EPFR6	SCK0E2 to SCK0E0	Other than 100 *
	DREQ2	DDRG	DDRG0	0
		PFRG	PFRG0	0
3	DACK3	PFRG	PFRG4	1
		EPFR21	TIOA2E1, TIOA2E0	Other than 10 *
		EPFR7	SOUT1E1, SOUT1E0	Other than 10 *
	DEOP3	PFRG	PFRG5	1
	DREQ3	DDRG	DDRG3	0

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

■ External bus interface

To assign an external bus interface pin, disable all the other pin settings assigned to the pin.

- A pins

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
4. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
A00	PFR2	PFR20	1
	EPFR10	SOUT4E1, SOUT4E0	Other than 01 *
	EPFR24	TIOA8E1, TIOA8E0	Other than 01 *
A01	PFR2	PFR21	1
A02	PFR2	PFR22	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR10	SCK4E1, SCK4E0	Other than 01 *
	EPFR24	TIOA9E1, TIOA9E0	Other than 01 *
A03	PFR2	PFR23	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
A04	PFR2	PFR24	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR0	OUT0E2 to OUT0E0	Other than 001 *
	EPFR11	SOUT5E1, SOUT5E0	Other than 01 *
	EPFR25	TIOA10E1, TIOA10E0	Other than 01 *
A05	PFR2	PFR25	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR0	OUT1E2 to OUT1E0	Other than 001 *
A06	PFR2	PFR26	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR1	OUT2E2 to OUT2E0	Other than 001 *
	EPFR11	SCK5E1, SCK5E0	Other than 01 *
	EPFR25	TIOA11E1, TIOA11E0	Other than 01 *

Pin Name	Register Name	Bit Name	Written Value
A07	PFR2	PFR27	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR1	OUT3E2 to OUT3E0	Other than 001 *
A08	PFR3	PFR30	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR12	SOUT6E1, SOUT6E0	Other than 01 *
	EPFR26	TIOA12E1, TIOA12E0	Other than 01 *
A09	PFR3	PFR31	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
A10	PFR3	PFR32	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR12	SCK6E1, SCK6E0	Other than 01 *
	EPFR26	TIOA13E1, TIOA13E0	Other than 01 *
A11	PFR3	PFR33	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
A12	PFR3	PFR34	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR2	OUT4E2 to OUT4E0	Other than 001 *
	EPFR13	SOUT7E1, SOUT7E0	Other than 01 *
	EPFR27	TIOA14E1, TIOA14E0	Other than 01 *
A13	PFR3	PFR35	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR2	OUT5E2 to OUT5E0	Other than 001 *
A14	PFR3	PFR36	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR3	OUT6E2 to OUT6E0	Other than 001 *
	EPFR13	SCK7E1, SCK7E0	Other than 01 *
	EPFR27	TIOA15E1, TIOA15E0	Other than 01 *

Pin Name	Register Name	Bit Name	Written Value
A15	PFR3	PFR37	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR3	OUT7E2 to OUT7E0	Other than 001 *
A16	PFR4	PFR40	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR14	SOUT8E1, SOUT8E0	Other than 01 *
A17	PFR4	PFR41	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
A18	PFR4	PFR42	1
	EPFR14	SCK8E1, SCK8E0	Other than 01 *
A19	PFR4	PFR43	1
A20	PFR4	PFR44	1
	EPFR15	SOUT9E1, SOUT9E0	Other than 01 *
A21	PFR4	PFR45	1
A22	PFR4	PFR46	1
	EPFR35	SIFE1, SIFE0	Other than 10 *
	EPFR15	SCK9E1, SCK9E0	Other than 01 *
A23	PFR4	PFR47	1
	EPFR35	SIFE1, SIFE0	Other than 10 *

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{AS}$ pin

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
4. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
$\overline{AS}$	PFR5	PFR54	1
	EPFR17	SOUT11E1, SOUT11E0	Other than 01 *

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{CS}$ pins

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
4. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
$\overline{CS0}$	PFR5	PFR50	1
	EPFR16	SOUT10E1, SOUT10E0	Other than 01 *
$\overline{CS1}$	PFR5	PFR51	1
$\overline{CS2}$	PFR5	PFR52	1
	EPFR16	SCK10E1, SCK10E0	Other than 01 *
$\overline{CS3}$	PFR5	PFR53	1

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- D pins

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
4. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
D00	PFR0	PFR00	1
	EPFR6	SOUT0E2 to SOUT0E0	Other than 010*
	EPFR20	TIOA0E1, TIOA0E0	Other than 01 *
D01	PFR0	PFR01	1
D02	PFR0	PFR02	1
	EPFR6	SCK0E2 to SCK0E0	Other than 010 *
	EPFR20	TIOA1E1, TIOA1E0	Other than 01 *
D03	PFR0	PFR03	1
D04	PFR0	PFR04	1
	EPFR7	SOUT1E1, SOUT1E0	Other than 01 *
	EPFR21	TIOA2E1, TIOA2E0	Other than 01 *
D05	PFR0	PFR05	1

Pin Name	Register Name	Bit Name	Written Value
D06	PFR0	PFR06	1
	EPFR7	SCK1E1, SCK1E0	Other than 01 *
	EPFR21	TIOA3E1, TIOA3E0	Other than 01 *
D07	PFR0	PFR07	1
D08	PFR1	PFR10	1
	EPFR8	SOUT2E1, SOUT2E0	Other than 01 *
	EPFR22	TIOA4E1, TIOA4E0	Other than 01 *
D09	PFR1	PFR11	1
D10	PFR1	PFR12	1
	EPFR8	SCK2E1, SCK2E0	Other than 01 *
	EPFR22	TIOA5E1, TIOA5E0	Other than 01 *
D11	PFR1	PFR13	1
D12	PFR1	PFR14	1
	EPFR9	SOUT3E1, SOUT3E0	Other than 01 *
	EPFR23	TIOA6E1, TIOA6E0	Other than 01 *
D13	PFR1	PFR15	1
D14	PFR1	PFR16	1
	EPFR9	SCK3E1, SCK3E0	Other than 01 *
	EPFR23	TIOA7E1, TIOA7E0	Other than 01 *
D15	PFR1	PFR17	1

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{RD}$ pin

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
3. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
$\overline{RD}$	PFR5	PFR55	1

- RDY pins

1. Set the port inputs on the DDR register (DDR=0).
2. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").

Pin Name	Register Name	Bit Name	Written Value
RDY	DDR6	DDR60	0

- SYSCLK pin

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
3. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
SYSCLK	PFR6	PFR61	1

- $\overline{WR0}$ and $\overline{WR1}$ pins

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Set an external bus interface (For details, see "CHAPTER 13 External Bus Interface").
4. Set peripheral functions on the PFR register (PFR=1).

Pin Name	Register Name	Bit Name	Written Value
$\overline{WR0}$	PFR5	PFR56	1
	EPFR17	SCK11E1, SCK11E0	Other than 01 *
$\overline{WR1}$	PFR5	PFR57	1

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

■ Slave interface

One of either of the SIFx or SIFx_1 pins can be selected for use with each channel.

To use the SIFA, $\overline{SIFCS}$, $\overline{SIFRD}$ and $\overline{SIFWR}$ pins, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Select a pin (port number) to be used on the EPFR register.
3. Disable the input of analog signal on the ADCHE register (ADE=0).
4. Enable the operation of slave interface (For details, see "CHAPTER 14 Slave Interface").

To use the SIFD, $\overline{SIFDRQR}$ and $\overline{SIFDRQT}$, the following settings are required.

1. Set the port inputs on the DDR register and the PFR register (DDR=0, PFR=0).
2. Disable the output of peripheral functions that share this pin with the EPFR register (For details of the multiplexed pins, see the pin assignment diagram).
3. Select a pin (port number) to be used on the EPFR register.
4. Set peripheral functions on the PFR register (PFR=1).

- SIFA pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	SIFA	DDRA	DDRA3	0
		PFRA	PFRA3	0
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE19	0
Port 1	SIFA_1	DDR4	DDR43	0
		PFR4	PFR43	0
		EPFR35	SIFE1, SIFE0	10

- $\overline{\text{SIFCS}}$ pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	$\overline{\text{SIFCS}}$	DDRA	DDRA2	0
		PFRA	PFRA2	0
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE18	0
Port 1	$\overline{\text{SIFCS\_1}}$	DDR4	DDR42	0
		PFR4	PFR42	0
		EPFR35	SIFE1, SIFE0	10

- SIFD pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	SIFD0	PFR7	PFR72	1
		EPFR35	SIFE1, SIFE0	01
		EPFR1	OUT2E2 to OUT2E0	Other than 010 *
		EPFR33	TMO0E1, TMO0E0	Other than 01 *
		ADCHE	ADE2	0
Port 1	SIFD0_1	PFR2	PFR22	1
		EPFR35	SIFE1, SIFE0	10
		EPFR24	TIOA9E1, TIOA9E0	Other than 01 *
		EPFR10	SCK4E1, SCK4E0	Other than 01 *

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	SIFD1	PFR7	PFR73	1
		EPFR35	SIFE1, SIFE0	01
		EPFR1	OUT3E2 to OUT3E0	Other than 010 *
		EPFR33	TMO1E1, TMO1E0	Other than 01 *
		ADCHE	ADE3	0
Port 1	SIFD1_1	PFR2	PFR23	1
		EPFR35	SIFE1, SIFE0	10
Port 0	SIFD2	PFR7	PFR74	1
		EPFR35	SIFE1, SIFE0	01
		EPFR2	OUT4E2 to OUT4E0	Other than 010 *
		EPFR34	TMO2E1, TMO2E0	Other than 01 *
		ADCHE	ADE4	0
Port 1	SIFD2_1	PFR2	PFR24	1
		EPFR35	SIFE1, SIFE0	10
		EPFR0	OUT0E2 to OUT0E0	Other than 001 *
		EPFR25	TIOA10E1, TIOA10E0	Other than 01 *
		EPFR11	SOUT5E1, SOUT5E0	Other than 01 *
Port 0	SIFD3	PFR7	PFR75	1
		EPFR35	SIFE1, SIFE0	01
		EPFR2	OUT5E2 to OUT5E0	Other than 010 *
		EPFR6	SOUT0E2 to SOUT0E0	Other than 001 *
		ADCHE	ADE5	0
Port 1	SIFD3_1	PFR2	PFR25	1
		EPFR35	SIFE1, SIFE0	10
		EPFR0	OUT1E2 to OUT1E0	Other than 001 *
Port 0	SIFD4	PFR7	PFR76	1
		EPFR35	SIFE1, SIFE0	01
		EPFR3	OUT6E2 to OUT6E0	Other than 010 *
		ADCHE	ADE6	0

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 1	SIFD4_1	PFR2	PFR26	1
		EPFR35	SIFE1, SIFE0	10
		EPFR1	OUT2E2 to OUT2E0	Other than 001 *
		EPFR25	TIOA11E1, TIOA11E0	Other than 01 *
		EPFR11	SCK5E1, SCK5E0	Other than 01 *
Port 0	SIFD5	PFR7	PFR77	1
		EPFR35	SIFE1, SIFE0	01
		EPFR3	OUT7E2 to OUT7E0	Other than 010 *
		EPFR6	SCK0E2 to SCK0E0	Other than 001 *
		ADCHE	ADE7	0
Port 1	SIFD5_1	PFR2	PFR27	1
		EPFR35	SIFE1, SIFE0	10
		EPFR1	OUT3E2 to OUT3E0	Other than 001 *
Port 0	SIFD6	PFR8	PFR80	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE8	0
Port 1	SIFD6_1	PFR3	PFR30	1
		EPFR35	SIFE1, SIFE0	10
		EPFR26	TIOA12E1, TIOA12E0	Other than 01 *
		EPFR12	SOUT6E1, SOUT6E0	Other than 01 *
Port 0	SIFD7	PFR8	PFR81	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE9	0
Port 1	SIFD7_1	PFR3	PFR31	1
		EPFR35	SIFE1, SIFE0	10
Port 0	SIFD8	PFR8	PFR82	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE10	0

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 1	SIFD8_1	PFR3	PFR32	1
		EPFR35	SIFE1, SIFE0	10
		EPFR26	TIOA13E1, TIOA13E0	Other than 01 *
		EPFR12	SCK6E1, SCK6E0	Other than 01 *
Port 0	SIFD9	PFR8	PFR83	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE11	0
Port 1	SIFD9_1	PFR3	PFR33	1
		EPFR35	SIFE1, SIFE0	10
Port 0	SIFD10	PFR8	PFR84	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE12	0
Port 1	SIFD10_1	PFR3	PFR34	1
		EPFR35	SIFE1, SIFE0	10
		EPFR2	OUT4E2 to OUT4E0	Other than 001 *
		EPFR27	TIOA14E1, TIOA14E0	Other than 01 *
		EPFR13	SOUT7E1, SOUT7E0	Other than 01 *
Port 0	SIFD11	PFR8	PFR85	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE13	0
Port 1	SIFD11_1	PFR3	PFR35	1
		EPFR35	SIFE1, SIFE0	10
		EPFR2	OUT5E2 to OUT5E0	Other than 001 *
Port 0	SIFD12	PFR8	PFR86	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE14	0
Port 1	SIFD12_1	PFR3	PFR36	1
		EPFR35	SIFE1, SIFE0	10
		EPFR3	OUT6E2 to OUT6E0	Other than 001 *
		EPFR27	TIOA15E1, TIOA15E0	Other than 01 *
		EPFR13	SCK7E1, SCK7E0	Other than 01 *

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	SIFD13	PFR8	PFR87	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE15	0
Port 1	SIFD13_1	PFR3	PFR37	1
		EPFR35	SIFE1, SIFE0	10
		EPFR3	OUT7E2 to OUT7E0	Other than 001 *
Port 0	SIFD14	PFRA	PFRA0	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE16	0
Port 1	SIFD14_1	PFR4	PFR40	1
		EPFR35	SIFE1, SIFE0	10
		EPFR14	SOUT8E1, SOUT8E0	Other than 01 *
Port 0	SIFD15	PFRA	PFRA1	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE17	0
Port 1	SIFD15_1	PFR4	PFR41	1
		EPFR35	SIFE1, SIFE0	10

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{\text{SIFDRQR}}$ pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	$\overline{\text{SIFDRQR}}$	PFRA	PFRA6	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE22	0
Port 1	$\overline{\text{SIFDRQR\_1}}$	PFR4	PFR46	1
		EPFR35	SIFE1, SIFE0	10
		EPFR15	SCK9E1, SCK9E0	Other than 01 *

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{\text{SIFDRQT}}$ pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	$\overline{\text{SIFDRQT}}$	PFRA	PFRA7	1
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE23	0
Port 1	$\overline{\text{SIFDRQT\_1}}$	PFR4	PFR47	1
		EPFR35	SIFE1, SIFE0	10

* : Do not write a setting prohibited value. For details, see "CHAPTER 15 I/O Ports".

- $\overline{\text{SIFRD}}$ pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	$\overline{\text{SIFRD}}$	DDRA	DDRA5	0
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE21	0
Port 1	$\overline{\text{SIFRD\_1}}$	DDR4	DDR45	0
		PFR4	PFR45	0
		EPFR35	SIFE1, SIFE0	10

- $\overline{\text{SIFWR}}$ pins

Port Number	Pin Name	Register Name	Bit Name	Written Value
Port 0	$\overline{\text{SIFWR}}$	DDRA	DDRA4	0
		PFRA	PFRA4	0
		EPFR35	SIFE1, SIFE0	01
		ADCHE	ADE20	0
Port 1	$\overline{\text{SIFWR\_1}}$	DDR4	DDR44	0
		PFR4	PFR44	0
		EPFR35	SIFE1, SIFE0	10

CHAPTER 3 CPU

This chapter explains the basics of the FR80 family CPUs, including its architecture, specifications, and instructions, to provide a better understanding of the CPU functions.

3.1 Memory Space
3.2 Features of the Internal Architecture
3.3 Operation Modes
3.4 Pipeline
3.5 Overview of Instructions
3.6 Basic Programming Model
3.7 Registers
3.8 Data Configuration
3.9 Addressing
3.10 Branch Instructions
3.11 EIT (Exception, Interrupt, Trap)

3.1 Memory Space

The logical address space of the FR80 family CPUs is 4 GB (2^{32} locations), and the CPUs can linearly access it.

■ Direct addressing areas

The address spaces $0000\ 0000_H$ to $0000\ 03FF_H$ are called the direct addressing areas.

These areas allow operands to be specified directly in instructions.

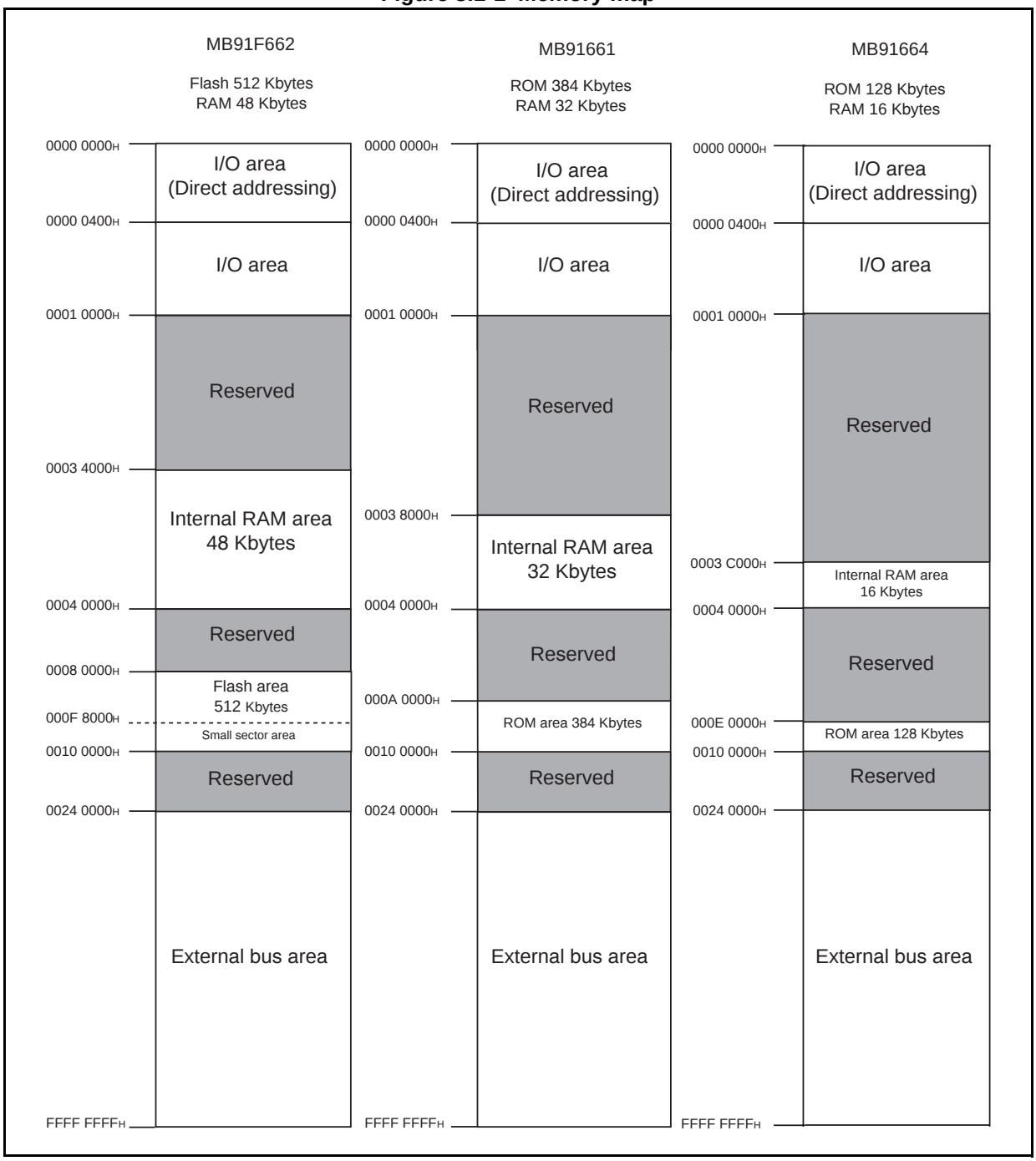
The direct addressing areas vary as follows depending on the size of the data accessed:

- Byte data access: $0000\ 0000_H$ to $0000\ 00FF_H$
- Half word data access: $0000\ 0000_H$ to $0000\ 01FF_H$
- Word data access: $0000\ 0000_H$ to $0000\ 03FF_H$

■ Memory map

Figure 3.1-1 shows a memory map of the MB91660 series.

Figure 3.1-1 Memory map



<Notes>

- For details of the small-sector area in flash memory, see "CHAPTER 36 Flash Memory". The small-sector area concerns only the flash memory products.
- Do not access the reserved areas.

3.2 Features of the Internal Architecture

The FR80 family CPUs have a high-performance core based on the RISC architecture with high-level functions and instructions included for embedded applications.

- Adoption of the RISC architecture
  Basic instructions: 1 instruction/1 cycle
- 32-bit architecture
  16 general-purpose 32-bit registers
- Linearly accessed 4-GB memory space
- Built-in multipliers
  - 32-bit × 32-bit multiplication: 5 cycles
  - 16-bit × 16-bit multiplication: 3 cycles
- Enhanced interrupt processing functions
  - High-speed response (6 cycles)
  - Multi-interrupt support
  - Level mask function (16 levels)
- Enhanced instructions for I/O operations
  - Memory-to-memory transfer instruction
  - Bit processing instruction
- High code efficiency
  - Basic instruction word length: 16 bits
- Compatibility of basic instructions with the FR60 family
- Addition of the following instructions to the instructions of the FR60 family:
  - Bit search instructions (SRCH0, SRCH1, and SRCHC)
- Deletion of the following instructions from the instructions of the FR60 family:
  - Coprocessor instructions (COPOP, COPLD, COPST, and COPSV)
  - Resource instructions (LDRES and STRES)
- Non-blocking load
  Up to 4 load instructions can be issued in advance.

3.3 Operation Modes

This section explains the operation modes of this series.

This series provides the operation modes below. At an activation of the device, one of these operation modes can be selected.

- User single-chip mode
- Serial programming mode

Table 3.3-1 lists the operation modes of this series.

Table 3.3-1 Operation modes

MD Pin		Control Pin	Operation Mode
MD1	MD0	P75	
0	0	X	User single-chip mode
	1	1	Serial programming mode

3.4 Pipeline

The FR architecture of the FR80 family CPUs is a compact 32-bit RISC architecture.
It has not only the normal instruction execution pipeline but also an additional pipeline for loading memory, which can reduce pipeline hazards during load instruction execution.

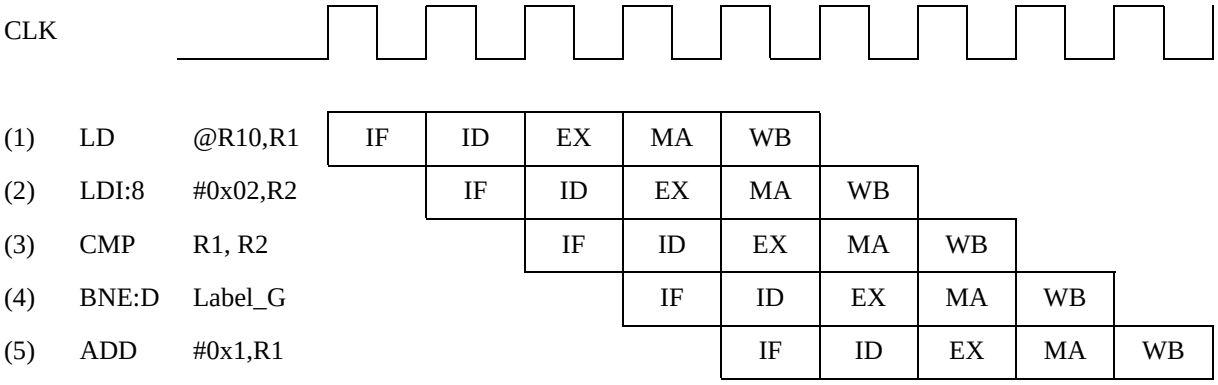
A five-stage instruction pipeline method is used in executing 1 instruction per cycle. The pipeline consists of the following stages:

- Instruction fetch (IF) stage: Fetches the instruction at the output address.
- Instruction decode (ID) stage: Decodes the fetched instruction. It also reads a register.
- Execution (EX) stage: Executes the decoded instruction.
- Memory access (MA) stage: Accesses the target memory.
- Register writing (WB) stage: Writes the operation results (or loaded memory data) to a register.

The pipeline for loading memory has been added so that the MA and WB stages of the instruction, which does not access memory, can overlap the MA and WB stages of an LD instruction.

As a rule, 1 instruction is executed per cycle. However, more than one cycle is required for execution of a load/store instruction with memory wait, a branch instruction without a delay slot, or a multi-cycle instruction. In addition, the instruction execution speed is slower when there is a delay in supplying an instruction.

Example 1:

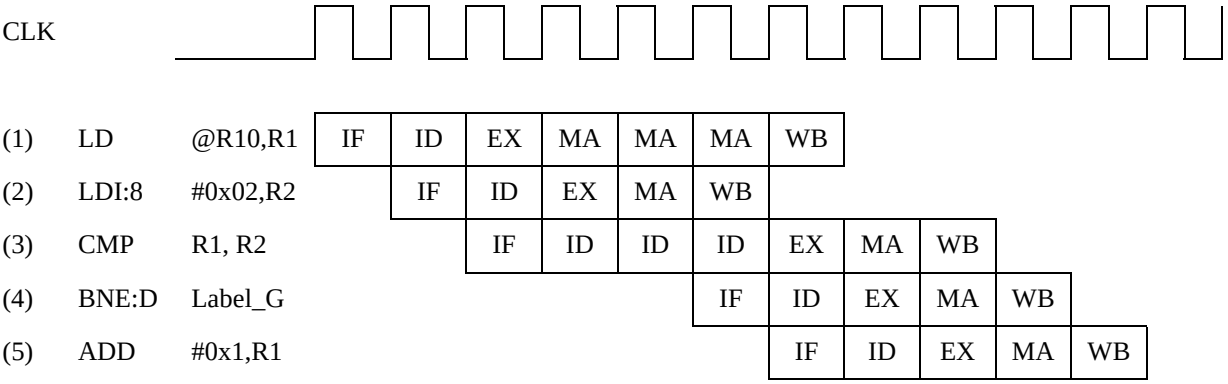


Example 1: The instructions are executed in sequence because the data that uses R1 to write the (1) LD instruction is returned in the (3) CMP instruction within 1 cycle.

In the load operation, the MA stage is extended until reading of the loaded data is completed.

However, if the register used for loading will not be used for the subsequent instructions, the instruction is executed as is.

Example 2:


CLK
(1) LD @R10,R1 IF ID EX MA MA MA WB
(2) LDI:8 #0x02,R2 IF ID EX MA WB
(3) CMP R1, R2 IF ID ID ID EX MA WB
(4) BNE:D Label_G IF ID EX MA WB
(5) ADD #0x1,R1 IF ID EX MA WB

Example 2: The data that uses R1 to write the (1) LD instruction is not returned within 1 cycle in the (3) CMP instruction, resulting in execution only up to the (2) LDI:8 instruction and keeping the CMP instruction waiting in the ID stage because of a register conflict.

3.5 Overview of Instructions

In addition to the general RISC instruction set, the FR80 family CPUs support the logical operations optimized for embedded applications, bit operation instructions, and direct addressing instructions. Each instruction has a length of 16 bits (some instructions have a length of 32 or 48 bits) and provides superior performance in memory usage efficiency.

The instruction sets can be divided into the following function groups:

- Arithmetic operation
- Load and store
- Branch
- Logical operation and bit operation
- Direct addressing
- Bit search
- Other

3.5.1 Arithmetic Operation

These instructions are standard arithmetic instructions (addition, subtraction, and comparison) and shift instructions (logical shift and arithmetic operation shift). The arithmetic operations of addition and subtraction can include operations with a carry used in individual operations with a multi-word length (operation for 32 or more bits of data) and operations suitable for address calculation in which flag values are not changed.

Also included in these instructions are the 32-bit × 32-bit multiplication instruction, 16-bit × 16-bit multiplication instruction, and 32-bit / 32-bit step division instruction.

The immediate transfer instruction that sets immediate data in a register and the register-to-register transfer instruction are also included.

All the operations of arithmetic operation instructions use the general-purpose registers and Multiply & Divide registers in the CPUs.

3.5.2 Load and Store

Load and store are instructions for reading and writing external memory. They are also used for reading and writing by the internal peripheral functions of the chip.

The access lengths of load and store are in any of 3 units: byte, half word, and word. In addition to general-purpose register indirect memory addressing, some load and store instructions can use register indirect memory addressing with either displacement or register increment/decrement operations.

3.5.3 Branch

Branch instructions include branch, call, interrupt, and return instructions. The branch instructions consist of instructions with delay slots and instructions without delay slots, and they can be optimized as required. For details of the branch instructions, see "3.10 Branch Instructions".

3.5.4 Logical Operation and Bit Operation

Logical operation instructions can perform the AND, OR, and EOR logical operations between general-purpose registers or between a general-purpose register and memory (and I/O). Also, bit operation instructions can directly manipulate data on memory (and of I/O).
Memory addressing is general-purpose register indirect memory addressing.

3.5.5 Direct Addressing

Direct addressing instructions are instructions used for access between I/O and a general-purpose register or between I/O and memory. Specifying an I/O address directly in an instruction instead of using register indirect addressing enables highly efficient high-speed access. Also, some direct addressing instructions can perform register indirect memory addressing with register increment/decrement operations.

3.5.6 Bit Search

A bit search instruction searches 32-bit data beginning from the MSB to obtain the bit location of the first "1" or "0" found in the register. A bit search instruction can also make a comparison with the MSB value and obtain the bit location of a value different from the first MSB found in a register.

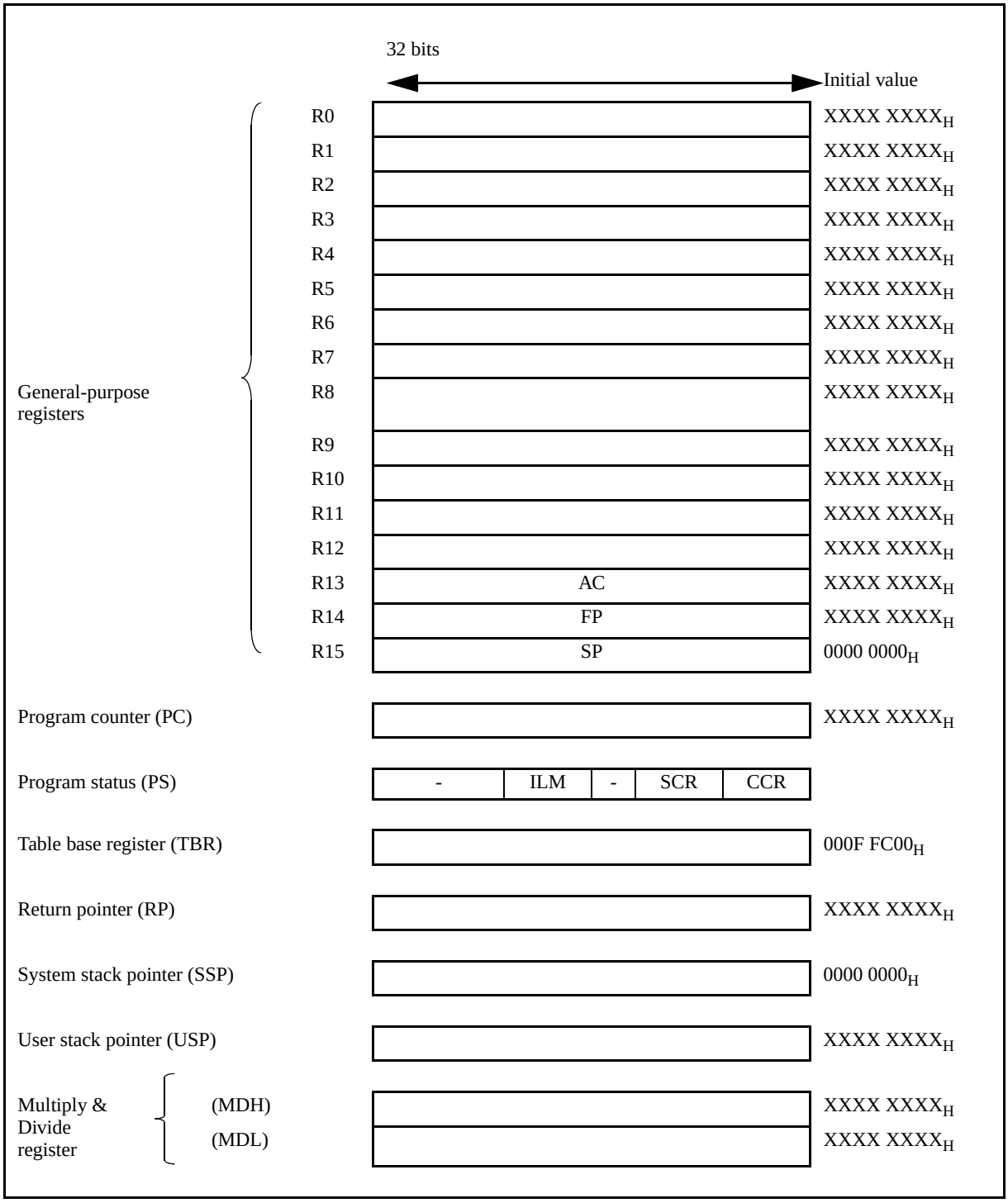
3.5.7 Other

Other available instructions include those for setting flags in the PS register, performing stack operations, and making a carry/zero extension. Also included in these instructions are function entry/exit instructions supporting high-level languages and multi-load/store instructions for registers.

3.6 Basic Programming Model

Figure 3.6-1 shows the basic programming model.

Figure 3.6-1 Basic programming model



3.7 Registers

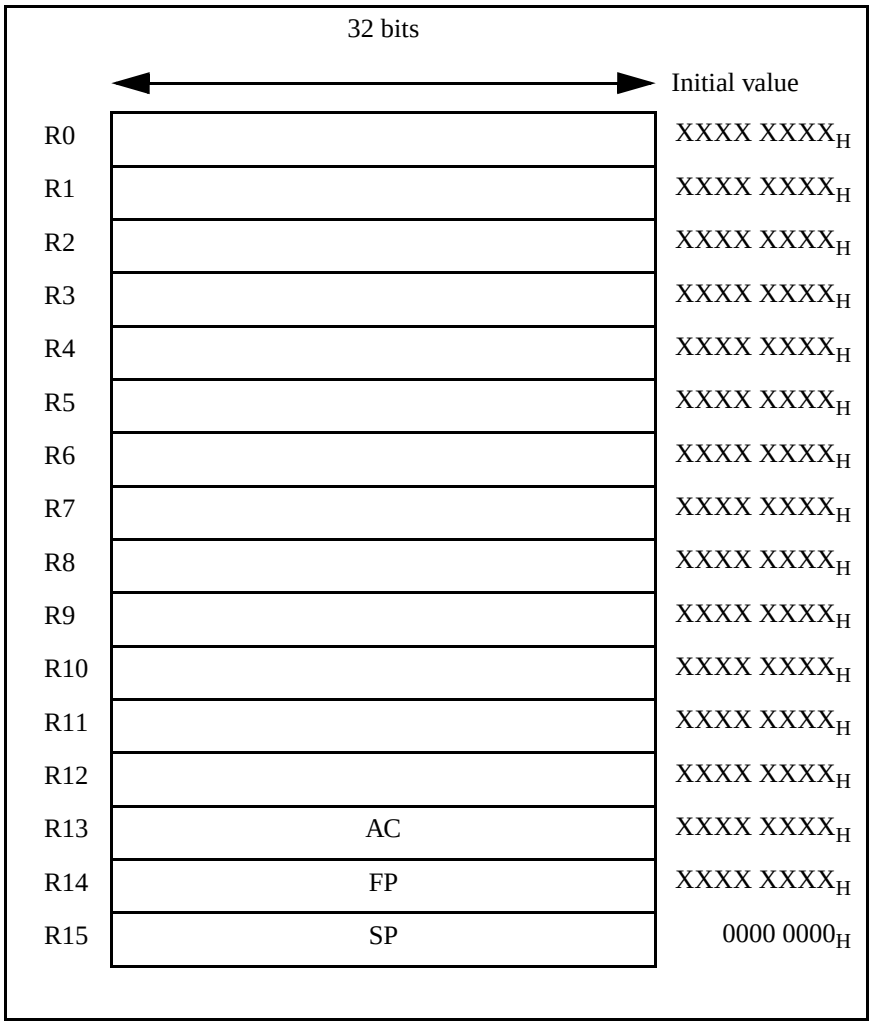
The register configuration consists of general-purpose registers and dedicated registers for specific purposes.

3.7.1 General-purpose Registers (R0 to R15)

Registers R0 to R15 are general-purpose registers. They are used as accumulators and memory access pointers in a variety of operations.

Figure 3.7-1 shows the bit configuration of the general-purpose registers (R0 to R15).

Figure 3.7-1 Bit configuration of the general-purpose registers (R0 to R15)



Of the 16 registers, the following registers are assumed to have specific purposes, and certain instructions have therefore been enhanced. For details of the initial values at the reset time, see Figure 3.7-1.

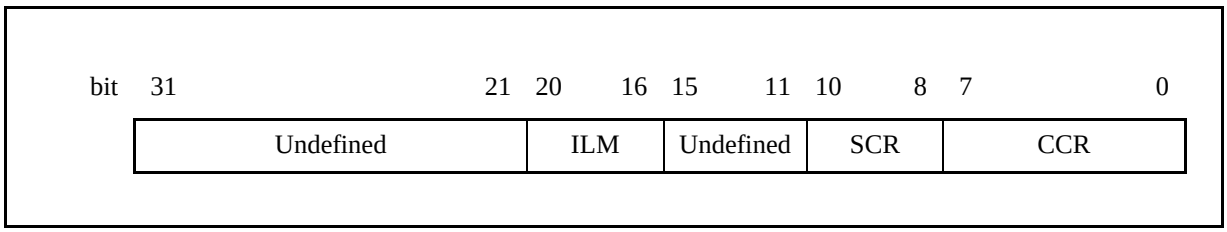
- R13: Virtual accumulator (AC)
- R14: Frame pointer (FP)
- R15: Stack pointer (SP)

3.7.2 Program Status Register (PS)

This register retains the program status, and it is divided into 3 parts: interrupt level mask register (ILM), system condition code register (SCR), and condition code register (CCR).

Figure 3.7-2 shows the bit configuration of the program status register (PS).

Figure 3.7-2 Bit configuration of the program status register (PS)



[bit31 to bit21, bit15 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is always read.

[bit20 to bit16] Interrupt level mask register (ILM)

See "■ Interrupt level mask register (ILM)".

[bit10 to bit8] System condition code register (SCR)

See "■ System condition register (SCR)".

[bit7 to bit0] Condition code register (CCR)

See "■ Condition code register (CCR)".

■ Condition code register (CCR)

Figure 3.7-3 shows the bit configuration of the condition code register (CCR).

Figure 3.7-3 Bit configuration of the condition code register (CCR)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	S	I	N	Z	V	C
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	X	X	X	X

R/W: Read/Write

-: Undefined

X: Undefined

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is always read.

[bit5]: S (Stack flag)

This bit specifies a stack pointer operating as general-purpose register 15 (R15).

S	Explanation
0	The system stack pointer (SSP) is operating as general-purpose register 15 (R15). The bit is automatically cleared to "0" when EIT occurs. (However, the value before the bit is cleared is saved to the stack.)
1	The user stack pointer (USP) is operating as general-purpose register 15 (R15).

This bit is cleared to "0" when the system is reset.

"0" must be written when the RETI instruction is executed.

[bit4]: I (Interrupt enable flag)

This bit controls enabling/disabling of user interrupt requests.

I	Explanation
0	Disables user interrupt requests. The bit is automatically cleared to "0" when the INT instruction is executed. (However, the value before the bit is cleared is saved to the stack.)
1	Enables user interrupt requests. The mask processing of user interrupt requests is controlled with the value retained by the interrupt level mask register (ILM).

This bit is cleared to "0" when the system is reset.

[bit3]: N (Negative flag)

This bit indicates a carry for an operation result recognized as an integer represented by a 2's complement.

N	Explanation
0	Indicates that the operation result is a positive value.
1	Indicates that the operation result is a negative value.

The initial state set by a reset is undefined.

[bit2]: Z (Zero flag)

This bit indicates whether the result of an operation is "0".

Z	Explanation
0	Indicates that the operation result is not "0".
1	Indicates that the operation result is "0".

The initial state set by a reset is undefined.

[bit1]: V (Overflow flag)

This bit indicates whether an overflow occurred as a result of an operation by interpreting each operand used for the operation as integers represented by 2's complements.

V	Explanation
0	No overflow occurred as a result of the operation.
1	An overflow occurred as a result of the operation.

The initial state set by a reset is undefined.

[bit0]: C (Carry flag)

This bit indicates whether a carry or borrow from the most significant bit occurred as a result of an operation.

C	Explanation
0	No carry or borrow occurred.
1	A carry or borrow occurred.

The initial state set by a reset is undefined.

■ System condition register (SCR)

Figure 3.7-4 shows the bit configuration of the system condition register (SCR).

Figure 3.7-4 Bit configuration of the system condition register (SCR)

bit	10	9	8
	D1	D0	T
Attribute	R/W	R/W	R/W
Initial value	X	X	0

R/W: Read/Write

X: Undefined

[bit10, bit9]: D1, D0 (Step division flag)

These bits retain in-process data during step division execution.

Do not change these bits while division processing is being executed.

To execute any other processing during step division, save and return the value of the program status register (PS). Doing so ensures a restart of step division.

The initial state set by a reset is undefined.

<Notes>

- The bits are set with the reference of the dividend and divisor by execution of the DIV0S instruction.
- They are forcibly cleared by execution of the DIV0U instruction.

[bit8]: T (Step trace trap flag)

This bit specifies whether the step trace trap is enabled.

T	Explanation
0	The step trace trap is disabled.
1	The step trace trap is enabled. All user interrupt requests are disabled.

This bit is cleared to "0" when the system is reset.

Emulators use the step trace trap function. The step trace trap cannot be used in a user program together with an emulator.

■ Interrupt level mask register (ILM)

This register retains the interrupt level mask value. The value retained by the register is used for the level mask.

Figure 3.7-5 shows the bit configuration of the interrupt level mask register (ILM).

Figure 3.7-5 Bit configuration of the interrupt level mask register (ILM)

bit	20	19	18	17	16
	ILM4	ILM3	ILM2	ILM1	ILM0
Attribute	R/W	R/W	R/W	R/W	R/W
Initial value	0	1	1	1	1

R/W: Read/Write

An interrupt request that is input to the CPU is accepted only if the corresponding interrupt level is higher than the level specified by this register.

The highest level is "0" (00000_B), and the lowest is "31" (11111_B).

A limited range of values can be set from programs.

- Original value in a range of 16 to 31: A value ranging from 16 to 31 can be specified as a new value. If a value ranging from 0 to 15 is set for an instruction, (specified-value + 16) is transferred when the instruction is executed.
- Original value in a range of 0 to 15: Any value ranging from 0 to 31 can be specified.

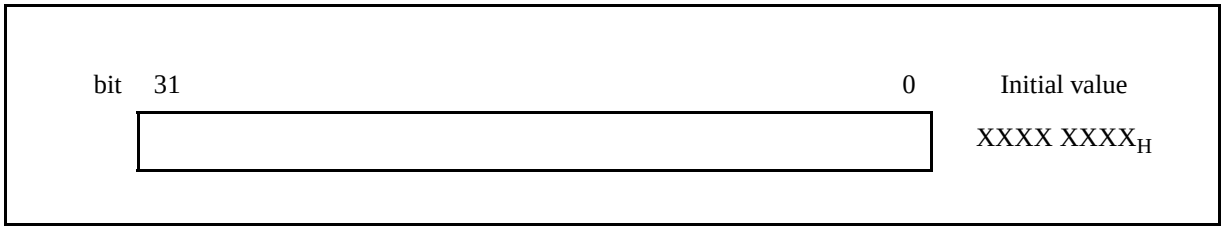
These bits are initialized to "15" (01111_B) by a reset.

3.7.3 Program Counter (PC)

This register is the program counter (PC) indicating the address of the instruction being executed.

Figure 3.7-6 shows the bit configuration of the program counter (PC).

Figure 3.7-6 Bit configuration of the program counter (PC)



bit0 is set to "0" when an instruction that entails a PC update is executed.

It is prohibited to specify an odd-numbered location as the branch destination address, and to set bit0 to "1".

The instruction would have to be located at an address that is a multiple of 2.

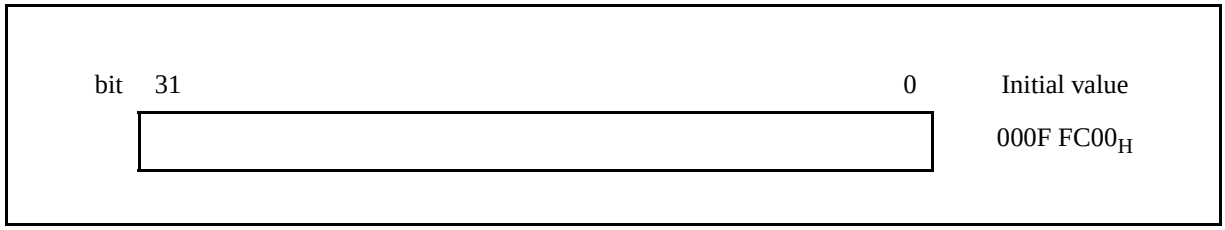
The initial value following a reset is undefined, and the program start address is set by a reset vector fetch.

3.7.4 Table Base Register (TBR)

This register retains the start address of the vector table used for EIT processing.

Figure 3.7-7 shows the bit configuration of the table base register (TBR).

Figure 3.7-7 Bit configuration of the table base register (TBR)



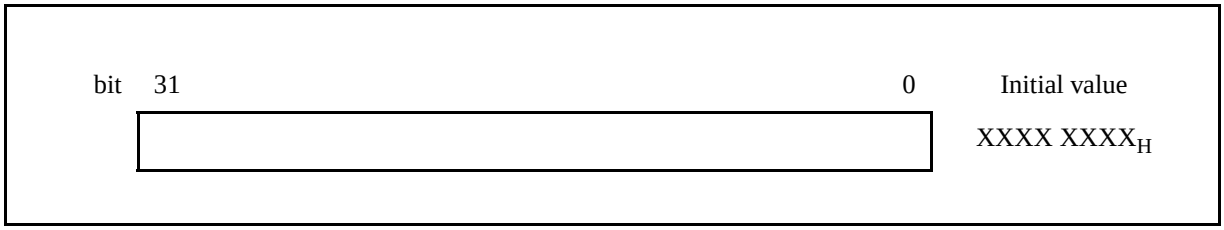
The initial value following a reset is "000F $FC00_H$".

3.7.5 Return Pointer (RP)

This pointer retains the return destination address when returning from a subroutine.

Figure 3.7-8 shows the bit configuration of the return pointer (RP).

Figure 3.7-8 Bit configuration of the return pointer (RP)



The value of the program counter (PC) is transferred to this register when the CALL instruction is executed.

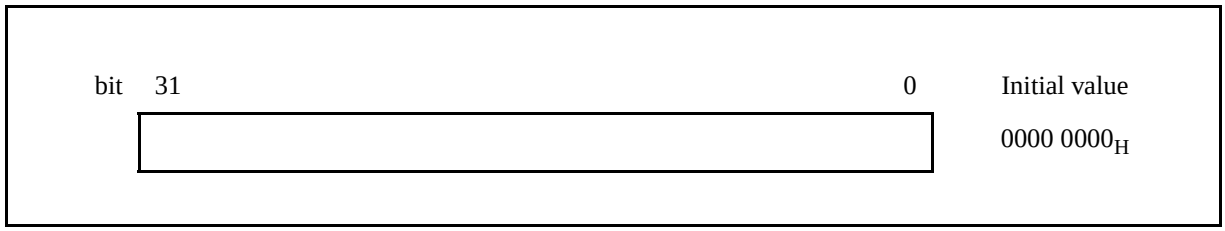
The register contents are transferred to the program counter (PC) when the RET instruction is executed.

3.7.6 System Stack Pointer (SSP)

This pointer operates as R15 when the S flag of the condition code register (CCR) is "0".
Also, the system stack pointer (SSP) can be specified explicitly.
It can be used as a stack pointer specifying the stack for saving the program status register (PS) and the program counter (PC) when EIT occurs.

Figure 3.7-9 shows the bit configuration of the system stack pointer (SSP).

Figure 3.7-9 Bit configuration of the system stack pointer (SSP)



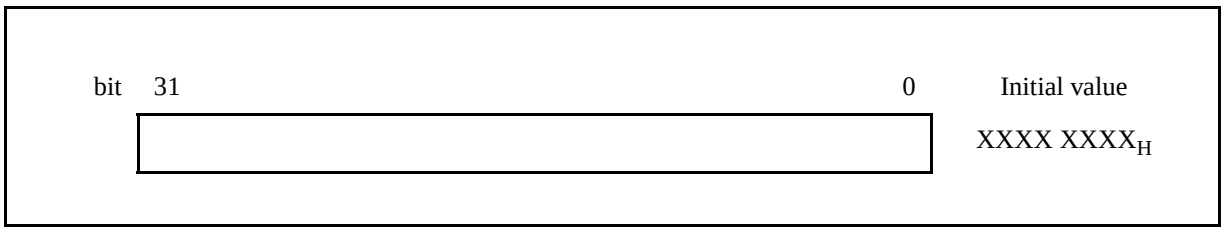
The initial value following a reset is "$0000\ 0000_H$".

3.7.7 User Stack Pointer (USP)

This pointer operates as R15 when the S flag of the condition code register (CCR) is "1". Also, the user stack pointer (USP) can be specified explicitly.

Figure 3.7-10 shows the bit configuration of the user stack pointer (USP).

Figure 3.7-10 Bit configuration of the user stack pointer (USP)



The initial value following a reset is undefined.

This pointer cannot be used in the RETI instruction.

3.7.8 Multiply & Divide Registers

These registers are used for multiplication and division, and each register has a length of 32 bits.

Figure 3.7-11 Bit configuration of the Multiply & Divide registers

The initial value following a reset is undefined.

● In multiplication

In multiplication of 32 bits × 32 bits, the result of an operation with a length of 64 bits is stored in the Multiply & Divide registers at the following locations:

- MDH: Upper 32 bits
- MDL: Lower 32 bits

In multiplication of 16 bits × 16 bits, the result is stored as follows:

- MDH: Undefined
- MDL: 32-bit result

● In division

The dividend is stored in MDL at the start of calculation.

In division according to the DIV0S, DIV0U, DIV1, DIV2, DIV3, or DIV4S instruction, the result is stored in MDH and MDL:

- MDH: Remainder
- MDL: Quotient

3.8 Data Configuration

Data is arranged in the FR80 family CPUs in the following two ways:

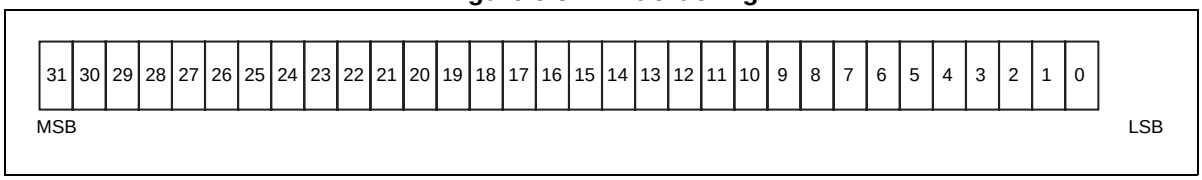
- Bit Ordering
- Byte Ordering

3.8.1 Bit Ordering

The FR80 family CPUs use little endian for bit ordering.

Figure 3.8-1 shows the bit ordering.

Figure 3.8-1 Bit ordering

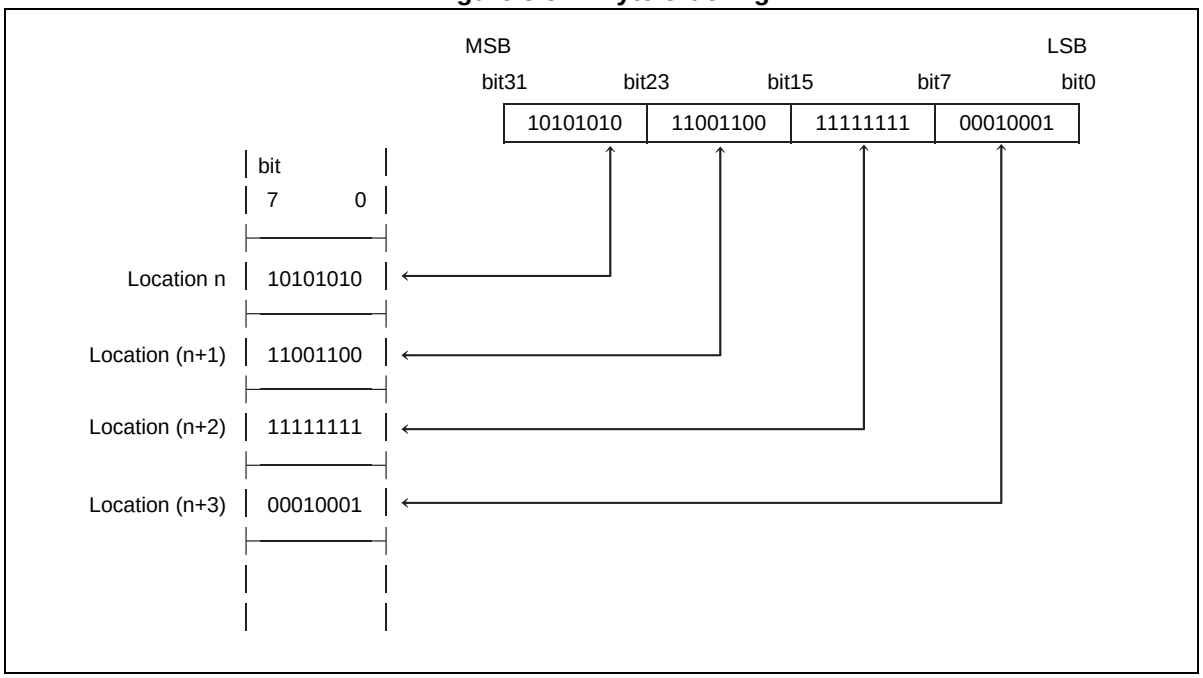


3.8.2 Byte Ordering

The FR80 family CPUs use big endian for byte ordering.

Figure 3.8-2 shows the byte ordering.

Figure 3.8-2 Byte ordering



3.8.3 Word Alignment

■ Program access

Programs for the FR80 family CPUs must be located at addresses that are multiples of 2. bit0 of the program counter (PC) is set to "0" when an instruction that entails the program counter (PC) update is executed. It is prohibited to specify an odd-numbered location as the branch destination address, and to set bit0 to "1".

The instruction would have to be located at an address that is a multiple of 2.

There is no odd-numbered address exception.

■ Data access

For an accessing of data in the FR80 family, set the address depending on the size of the data accessed as shown below. (The address is not aligned by the hardware.)

Word access: The address is a multiple of 4 (the lowest 2 bits are set to "00").

Half word access: The address is a multiple of 2 (the lowest bit is set to "0").

Byte access: ----

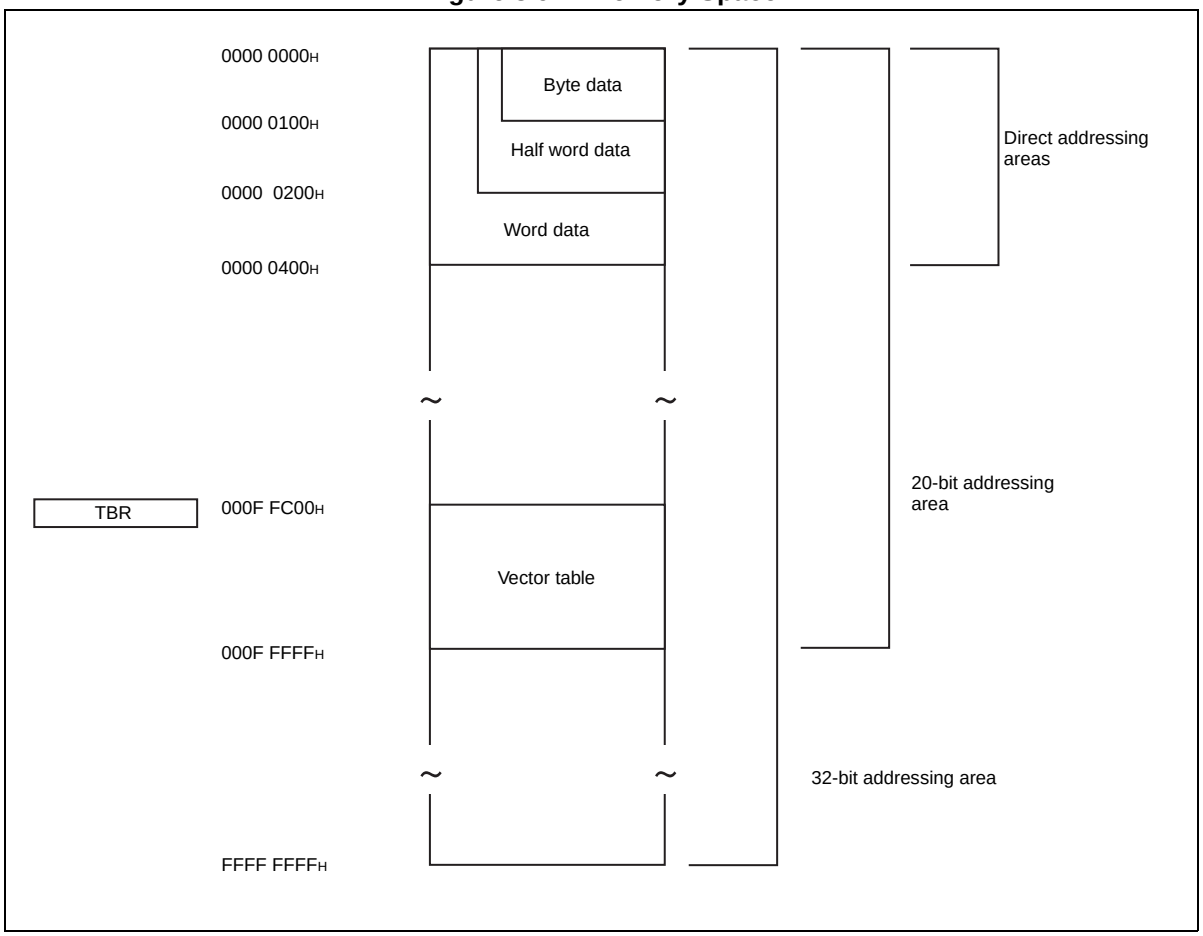
During a word or half word data access, set the above address for the result from a calculation of the effective address.

3.9 Addressing

The memory space consists of linear 32-bit addresses.

Figure 3.9-1 shows the memory space.

Figure 3.9-1 Memory Space



3.9.1 Direct Addressing Areas

The memory space areas listed below are areas for I/O. Direct addressing enables these areas to be specified directly as operand addresses in instructions.

The size of an address area that can be specified by a direct address varies depending on the data length.

- Byte data (8 bits): 0 to 0x0FF
- Half word data (16 bits): 0 to 0x1FF
- Word data (32 bits): 0 to 0x3FF

3.9.2 20-bit Addressing Area

20-bit addressing area: 0 to 0xFFFFF

If all the program and data areas are located in the 20-bit addressing area, programs will be more compact and therefore have high performance after compilation.

An example of expansion of a normal 20-bit branch macro instruction is shown below.

```
BRA20       label20,Ri
        ↓                   Code size
LDI:20      #label20,Ri     ;  4 bytes
JMP         @Ri             ;  2 bytes
                            ---------
                            Total: 6 bytes
```

For details, see the "FR FAMILY SOFTUNE C/C++ COMPILER MANUAL for V6".

3.9.3 32-bit Addressing Area

32-bit addressing area: 0 to 0xFFFFFFFF

If the program and data areas are located beyond the 20-bit addressing area, the code sizes of programs will be larger than those of programs created in the 20-bit addressing area.

An example of expansion of a normal 32-bit branch macro instruction is shown below.

```
BRA32       label32,Ri
        ↓                   Code size
LDI:32      #label32,Ri     ;  6 bytes
JMP         @Ri             ;  2 bytes
                            ---------
                            Total: 8 bytes
```

For details, see the "FR FAMILY SOFTUNE C/C++ COMPILER MANUAL for V6".

3.9.4 Vector Table Initial Area

The area from 000F $FC00_H$ to 000F $FFFF_H$ is the EIT vector table initial area.

The vector table used for EIT processing can be placed at an arbitrary address by changing the table base register (TBR) accordingly, but the initial address following a reset is the above address.

3.10 Branch Instructions

Operation with delay slots and operation without delay slots can be specified for branch instructions in the FR80 family CPUs.

3.10.1 Operation with Delay Slots

■ Instructions

The following instructions perform branch operations with delay slots:

JMP:D	@Ri	/	CALL:D	label12	/	CALL:D	@Ri	/	RET:D	
BRA:D	label9	/	BNO:D	label9	/	BEQ:D	label9	/	BNE:D	label9
BC:D	label9	/	BNC:D	label9	/	BN:D	label9	/	BP:D	label9
BV:D	label9	/	BNV:D	label9	/	BLT:D	label9	/	BGE:D	label9
BLE:D	label9	/	BGT:D	label9	/	BLS:D	label9	/	BHI:D	label9

■ Explanation of operation

The instruction that is located immediately following a branch instruction (the location is called a "delay slot") is executed before branching, and an instruction at the branch destination is executed after that. Because the instruction in the delay slot is executed before the branch operation, the apparent execution speed is 1 cycle. Such being the case, if no valid instruction can be entered in the delay slot, the NOP instruction must be placed there instead.

Example:

```
;          Order of instructions
           ADD       R1, R2;
           BRA:D     LABEL     ; Branch instruction
           MOV       R2, R3    ; Delay slot          ..... Executed before branching
           ...
LABEL:     ST        R3, @R4   ; Branch destination
```

The conditional branch instruction that is located in the delay slot is executed whether the branch condition is satisfied or not.

Although the sequence of execution of some instructions seems to be inverted for delay branch instructions, the sequence is inverted only when the program counter (PC) is updated. Any other operations, such as updating or referencing a register, are executed in the sequence described.

Concrete explanations are given below.

1. Ri referenced by the JMP:D @Ri / CALL:D @Ri instruction is not affected even when updated by the instruction in a delay slot.

   Example:

```
LDI:32   #Label, R0
JMP:D    @R0        ; Branching to Label
LDI:8    #0, R0     ; The branch destination address is not affected.
...
```

2. The return pointer (RP) referenced by the RET:D instruction is not affected even when the instruction in a delay slot updates the return pointer (RP).

   Example:

```
RET:D               ; Branching to the address indicated by the RP
                      specified beforehand
MOV      R8, RP     ; The return operation is not affected.
...
```

3. The flag referenced by the Bcc:D rel instruction is not affected by the instruction in a delay slot either.

   Example:

```
ADD      #1, R0     ; Flag change
BC:D     Overflow   ; Branching according to the execution result of the
                      above instruction
ANDCCR   #0         ; This flag update is not referenced in the above
                      branch instruction.
...
```

4. When the RP is referenced in an instruction in the delay slot of the CALL:D instruction, the updated contents are read by the CALL:D instruction.

   Example:

```
CALL:D   Label      ; RP update and branching
MOV      RP, R0     ; Transfer of the RP of the execution result for the
                      above CALL:D
...
```

■ Instructions that can be placed in delay slots

Only instructions that satisfy the following conditions can be executed in delay slots:

- 1-cycle instruction
- Not a branch instruction
- Instruction that does not affect operations even if the order of execution is changed

■ Step trace trap

No step trace trap occurs between execution of a branch instruction with a delay slot and the delay slot.

■ Interrupts

No interrupt is accepted between execution of a branch instruction with a delay slot and the delay slot.

■ Undefined instruction exception

If the instruction in a delay slot is undefined, no undefined instruction exception occurs. In such cases, the undefined instruction operates as the NOP instruction.

3.10.2 Operation without Delay Slots

■ Instructions

The following instruction performs branch operations without delay slots:

JMP	@Ri	/	CALL	label12	/	CALL	@Ri	/	RET	
BRA	label9	/	BNO	label9	/	BEQ	label9	/	BNE	label9
BC	label9	/	BNC	label9	/	BN	label9	/	BP	label9
BV	label9	/	BNV	label9	/	BLT	label9	/	BGE	label9
BLE	label9	/	BGT	label9	/	BLS	label9	/	BHI	label9

■ Explanation of operation

Instructions are executed in the order they are listed. No instruction that is coded immediately following a branch instruction is executed before branching.

Example:

```
;           Order of instructions
            ADD     R1, R2      ;
            BRA     LABEL       ; Branch instruction (without a delay slot)
            MOV     R2, R3      ; Not executed
            ...
LABEL       ST      R3, @R4     ; Branch destination
```

The number of execution cycles of a branch instruction without a delay slot is 2 cycles if there is branching and 1 cycle if there is no branching.

Such operation increases the instruction code efficiency compared with that of branch instructions with delay slots in which NOP is clearly written because appropriate instructions cannot be placed in the delay slots.

If valid instructions can be placed in delay slots, select operation with delay slots; otherwise, select operation without delay slots. Doing so can balance execution speed with code efficiency.

3.11 EIT (Exception, Interrupt, Trap)

EIT stands for Exception, Interrupt, and Trap. It indicates that the event that occurred results in suspension of execution of the current program, and the execution of another program.
An exception is an event that occurs in connection with the context being executed. The processing is reexecuted beginning with the instruction that causes an exception.
An interrupt is an event that occurs independently of the context being executed. The source of events is hardware.
A trap is an event that occurs in connection with the context being executed. Some traps occur as instructed in programs such as a system call. The instruction following the instruction that generates a trap is reexecuted first.

■ Features

- Multi-EIT support
- Level mask function for interrupts (A user can use 15 levels.)
- Trap instructions (INT/INTE)
- EIT for emulator activation (hardware/software)

3.11.1 EIT Sources

EIT sources include the following:

- Reset
- User interrupt (peripheral functions, external interrupts)
- Delay interrupt
- Undefined instruction exception
- Trap instruction (INT)
- Trap instruction (INTE)
- Step trace trap

3.11.2 Return from EIT

The return from each EIT is through the RETI instruction.

3.11.3 Interrupt Level

The interrupt levels are 0 to 31, and they are controlled in units of 5 bits.

Table 3.11-1 lists the assignment of each level.

Table 3.11-1 Interrupt level assignment table

Level		Interrupt Type	Remarks
Binary number	Decimal number		
00000	0	(Reserved for system)	If the original value of the interrupt level mask register (ILM) is in a range of 16 to 31, no value in this range can be specified for the interrupt level mask register (ILM) from the program.
. . .	. . .	. . .	
. . .	. . .	. . .	
00011	3	(Reserved for system)	
00100	4	INTE instruction Step trace trap	
00101	5	(Reserved for system)	
. . .	. . .	. . .	
. . .	. . .	. . .	
01100	14	(Reserved for system)	
01101	15	(Reserved for system)	
10000	16	Interrupt request	When the interrupt level mask register (ILM) is set, user interrupts must be disabled.
10001	17	Interrupt request	
. . .	. . .	. . .	
. . .	. . .	. . .	
11110	30	Interrupt request	
11111	31	-	If the interrupt control register (ICR) is set, interrupts are disabled.

The operations are enabled only if the level is in a range of 16 to 31.

The interrupt level does not affect undefined instruction exceptions and the INT instruction. It does not change the interrupt level mask register (ILM) either.

3.11.4 I Flag

This flag specifies whether interrupts are enabled or disabled. It is provided as bit4 of the condition code register (CCR) in the program status register (PS).

I	Explanation
0	The bit is automatically cleared to "0" when the INT instruction is executed. (However, the value that is saved to the stack is that immediately before the bit is cleared.)
1	The mask processing of user interrupt requests is controlled with the value retained by the interrupt level mask register (ILM).

<Note>

After an instruction changes the value of the I flag, interrupt requests can be accepted beginning from the instruction after the next instruction.

Therefore, to operate interrupts properly, NOP must be placed after the instruction that changes the I flag value.

- Enabling interrupts (I flag = 1)

Instruction execution		I flag	Interrupts	
↓	ORCCR #set_iflag	0	Disabled	
	NOP	1	Disabled	
	Instruction A	1	Enabled	↑ **Starts enabling interrupts**

- Disabling interrupts (I flag = 0)

Instruction execution		I flag	Interrupts	
↓	ANDCCR #clear_iflag	1	Enabled	
	NOP	0	Enabled	
	Instruction A	0	Disabled	↑ **Starts disabling interrupts**

3.11.5 Interrupt Level Mask Register (ILM)

This register retains the interrupt level mask value. The register is provided as bit20 to bit16 of the program status register (PS).

An interrupt request that is input to a CPU in the FR80 family CPUs is accepted only if the corresponding interrupt level is higher than the level specified by the interrupt level mask register (ILM).

The highest level is "0" (00000), and the lowest is "31" (11111).

A limited range of values can be set from programs. If the original value is in a range of 16 to 31, a value ranging from 16 to 31 can be specified as a new value. If a value ranging from 0 to 15 is set for an instruction, (specified-value + 16) is transferred when the instruction is executed.

If the original value is in a range of 0 to 15, any value ranging from 0 to 31 can be specified. Use the STILM instruction for this setting.

<Note>

After an instruction changes the value of the interrupt level mask register (ILM), interrupt requests can be accepted beginning from the instruction after the next instruction.

Therefore, to operate interrupts properly, NOP must be placed after the instruction that changes the interrupt level mask register (ILM).

Instruction execution		ILM	Interrupt Accepted	
↓	SETILM #set_ILM_B	A	A	
	NOP	B	A	
	Instruction C	B	B	↑
	Instruction D	B	B	**Starts enabling ILM=B.**

3.11.6 Level Mask for Interrupts

When an interrupt request is generated, the interrupt level of the interrupt source is compared with the level mask value retained by the interrupt level mask register (ILM). Then, if the following condition is satisfied, the source is masked and the request is not accepted:

Interrupt level of source ≥ Level mask value

3.11.7 Interrupt Control Register (ICR)

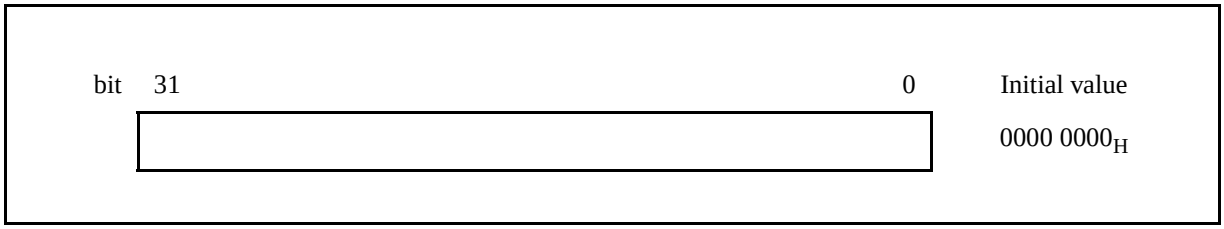
See "CHAPTER 10 Interrupt Controller".

3.11.8 System Stack Pointer (SSP)

This pointer indicates the stack used for saving or restoring data, when EIT has been received or the return operation is performed.

Figure 3.11-1 shows the bit configuration of the system stack pointer (SSP).

Figure 3.11-1 Bit configuration of the system stack pointer (SSP)



"8" is subtracted during EIT processing, and "8" is added at the time of return from EIT with the RETI instruction executed.

The initial value following a reset is "0000 0000_H".

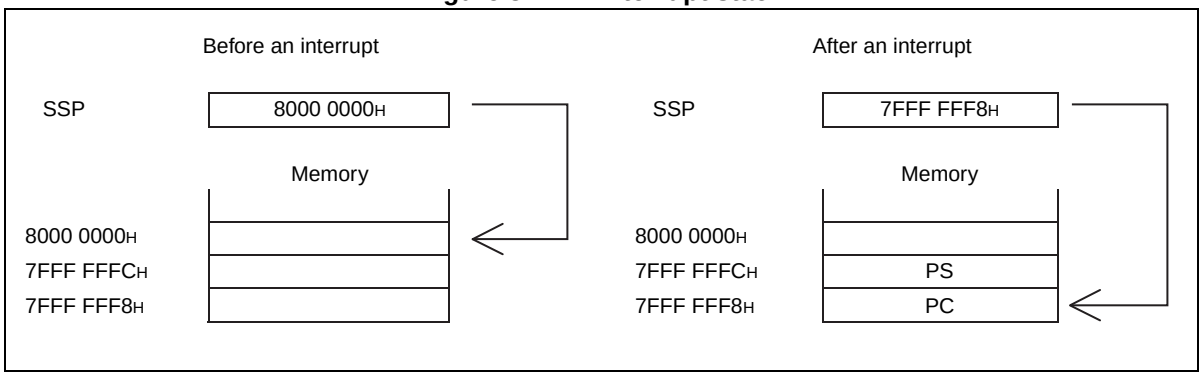
This pointer operates as general-purpose register R15 when the S flag of the condition code register (CCR) is "0".

3.11.9 Interrupt Stack

The interrupt stack is the area specified by the system stack pointer (SSP). It saves and restores the values of the program counter (PC) and the program status register (PS). After an interrupt, the value of the program counter (PC) is stored in the address specified by the system stack pointer (SSP), and the value of the program status register (PS) is stored in the address specified by the system stack pointer (SSP) plus 4.

Figure 3.11-2 shows the interrupt stack.

Figure 3.11-2 Interrupt stack

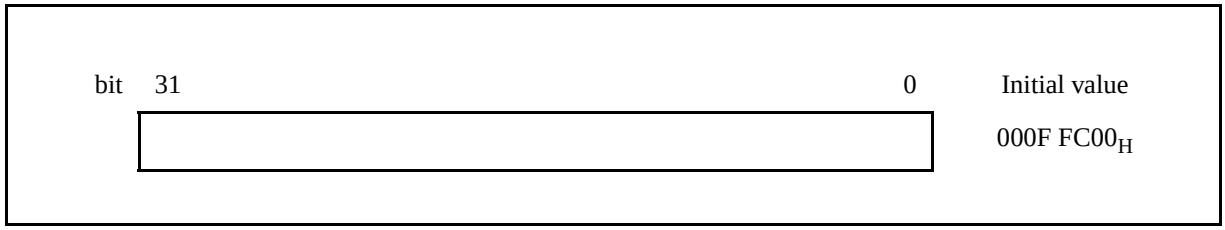


3.11.10 Table Base Register (TBR)

This register indicates the start address of the vector table used for EIT processing.

Figure 3.11-3 shows the bit configuration of the table base register (TBR).

Figure 3.11-3 Bit configuration of the table base register (TBR)



A vector address is the table base register (TBR) value plus the offset value assigned to each EIT source.

The initial value following a reset is "000F $FC00_H$".

3.11.11 EIT Vector Table

The vector area for EIT processing is the 1-KB area from the address specified by the table base register (TBR).

The size of 1 vector is 4 bytes, and the relationship between interrupt vector numbers and vector addresses is expressed as follows:

vctadr = TBR + vctofs

= TBR + (0x3FC - 4 × vct)

vctadr: Vector address vctofs: Vector offset vct: Interrupt vector number

TBR: Table base register

The lowest 2 bits of the addition result are always handled as "00".

The initial area of the vector table following a reset is the area from 000F $FC00_H$ to 000F $FFFF_H$.

Specific functions are assigned to part of the vectors.

3.11.12 Multi-EIT Processing

If multiple EIT sources occur at one time, the CPU selectively selects and accepts 1 EIT source, executes the EIT sequence, detects EIT sources again, and then repeats these actions. When no more detected EIT sources can be accepted, the CPU executes the handler instruction of the last EIT source accepted.

Therefore, if multiple EIT sources occur at one time, the sequence in which the handler of each source is executed depends on the following:

1. Priority in which EIT sources are accepted
2. The mask applied to other sources after a source is accepted

The sequence of execution depends on the above 2 elements.

The priority in which EIT sources are accepted is the order of selection of the source whose EIT sequence will be executed. In the EIT sequence, the program status register (PS) and the program counter (PC) are saved, the program counter (PC) is updated, and the other sources are masked as required. The handler of a source accepted earlier is not necessarily executed earlier.

Table 3.11-2 outlines the priority in which EIT sources are accepted.

Table 3.11-2 Priority in which EIT sources are accepted and masking of other sources

Priority of Acceptance	Source	Masking of Other Sources	ILM
1	Reset	The other sources are abandoned.	15
2	Other than undefined instructions	All sources of lower priority	-
3	INT instruction	I flag = 0	-
4	INTE instruction	All sources of lower priority	4
5	User interrupt	ILM = Level of accepted source	ICR
6	Step trace trap	All sources of lower priority	4

With additional consideration given to the masking of other sources after an EIT source is accepted, the sequence of execution of the handlers of EIT sources that occur at one time is as shown below.

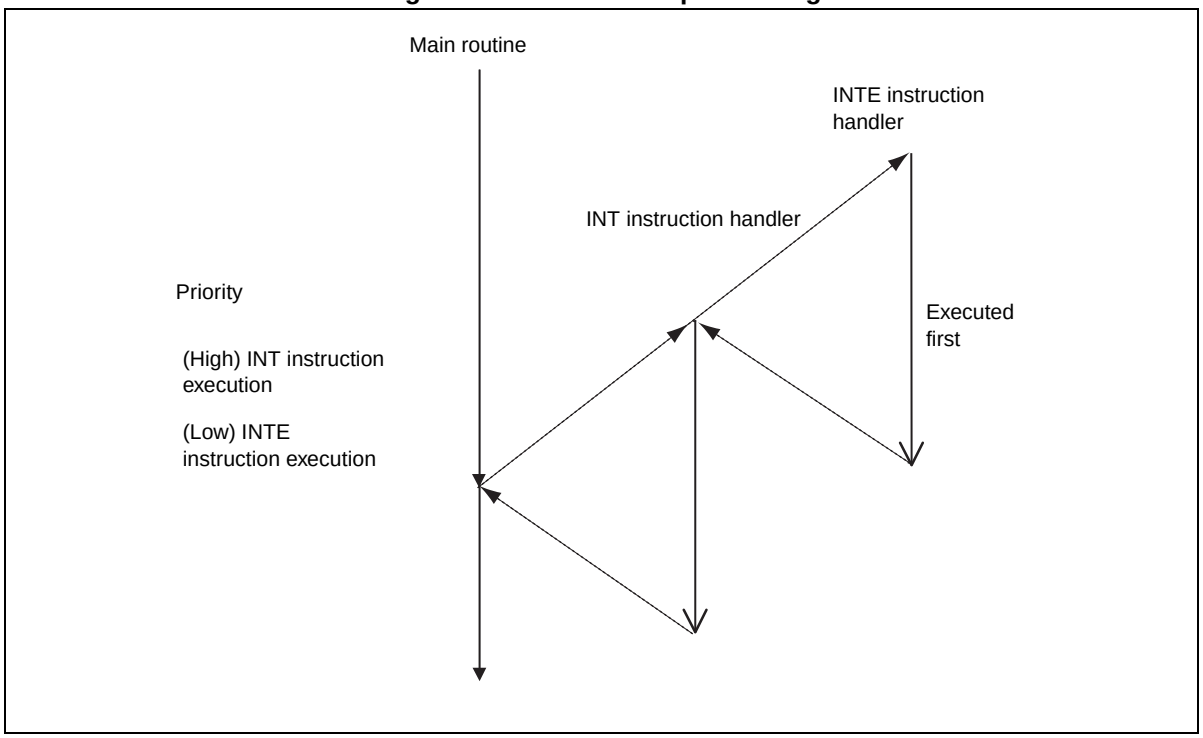
Table 3.11-3 lists the sequence of execution.

Table 3.11-3 Sequence of EIT handler execution

Priority of Acceptance	Source
1	Reset
2	Other than undefined instructions
3	INTE instruction
4	Step trace trap
5	INT instruction
6	User interrupt

Figure 3.11-4 shows multi-EIT processing.

Figure 3.11-4 Multi-EIT processing



3.11.13 Operation

In the explanations in this section, the PC of the transfer source indicates the address of the instruction that detects each EIT source.

"Next instruction address" indicates the value corresponding to the case where each of the instructions below that detects EIT satisfies the respective condition shown:

- For LDI:32 instruction: PC + 6
- For LDI:20 instruction: PC + 4
- For other instructions: PC + 2

■ User interrupt operation

The sequence in which a generated user interrupt request is determined as accepted or not is shown below.

User interrupt requests are generated from peripheral functions, and an interrupt level is set for every interrupt request.

● Acceptance of interrupt requests

1. The levels of interrupt requests generated simultaneously are compared, and the interrupt with the highest level (with the lowest numerical value) is selected.

   The value retained by the corresponding interrupt control register (ICR) is used for this comparison.
2. If multiple interrupt requests generated at one time have the same interrupt level, the interrupt request with the lowest interrupt number is selected.
3. An interrupt request with an interrupt level greater than or equal to the level mask value is masked and not accepted.

   If the level mask value is greater than the interrupt level, go to 4.

4. In cases where the selected interrupt request can be masked, if the I flag is "0", the interrupt request is masked and not accepted. If the I flag is "1", the interrupt request is accepted.

   Under the above conditions, the interrupt request will be accepted when one instruction processing is completed.

When an instruction that changes the I flag or interrupt level mask register (ILM) is executed, EIT control with the new acceptance condition becomes effective after 2 instructions.

If an EIT request is detected at the same time that a user interrupt request is accepted, the CPU operates as follows using the interrupt number corresponding to the accepted interrupt request.

* The parentheses () in "● Operation" below indicate the address that a register points to.

● Operation

1	(TBR + vector offset of the accepted interrupt request)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	Next instruction address	→ (SSP)
6	Interrupt level of the accepted request	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

After the interrupt sequence is completed, detection of any new EIT is performed before the first instruction of the handler is executed. If any EIT that occurred can be accepted at this point, the CPU switches to the EIT processing sequence.

3.11.14 INT Instruction Operation

The INT #u8 instruction generates a trap in software.
It generates a trap with the interrupt number specified in the operand.

● Operation

1	(TBR + 0x3FC - 4 × u8)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC + 2	→ (SSP)
6	"0"	→ I flag
7	"0"	→ S flag
8	TMP	→ PC

3.11.15 INTE Instruction Operation

The INTE instruction generates a trap in software for debugging.

● Operation

1	(TBR + 0x3D8)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC + 2	→ (SSP)
6	"00100_B"	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

3.11.16 Step Trace Trap Operation

The step trace trap is a trap for debugging, and it is generated for each single instruction execution by setting the T flag of the program status register (PS). No step trace trap is generated immediately after execution of a branch instruction during execution of a delay branch instruction. It is generated after the instruction in the delay slot is executed.

● Step trace trap detection conditions

1. T flag of the program status register (PS) = 1
2. The instruction being executed is not a delay branch instruction.
3. The CPU is in user mode.

   If the above conditions are satisfied, a break is set when one instruction operation processing is completed.

● Operation

1	(TBR + 0x3C4)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	Next instruction address	→ (SSP)
6	"00100_B"	→ ILM
7	"0"	→ S flag
8	TMP	→ PC

If the T flag = 1, user interrupts are disabled.

3.11.17 Undefined Instruction Exception Operation

When the instruction being decoded is detected as being undefined, an undefined instruction exception is generated.

● Undefined instruction exception detection conditions

1. The instruction being decoded is detected as being undefined.
2. The instruction is not in a delay slot (i.e., it does not immediately follow a delay branch instruction).

If the above conditions are satisfied, an undefined instruction exception is generated and a break is set.

● Operation

1	(TBR + 0x3C4)	→ TMP
2	SSP - 4	→ SSP
3	PS	→ (SSP)
4	SSP - 4	→ SSP
5	PC	→ (SSP)
6	"0"	→ S flag
7	TMP	→ PC

The address of the instruction that detects an undefined instruction exception is saved as the program counter (PC).

3.11.18 RETI Instruction Operation

The RETI is an instruction to return from the EIT processing routine.

● Operation

1	(R15)	→ PC
2	R15 + 4	→ R15
3	(R15)	→ PS
4	R15 + 4	→ R15

The S flag must be "0" when the RETI instruction is executed.

3.11.19 Delay Slots and EIT

The delay slots of branch instructions have the following restrictions concerning EIT.

● Interrupts, traps

No interrupt or trap occurs between execution of a branch instruction with a delay slot and the delay slot.

● Exceptions

If the instruction in a delay slot is undefined, no undefined instruction exception occurs. In such cases, the undefined instruction operates as the NOP instruction.

CHAPTER 4 Clock Generating Parts

This chapter explains the clock generating parts that generate the source clock (SRCCLK), which is the source of all internal clocks in this device.

4.1 Overview
4.2 Configuration
4.3 Pins
4.4 Registers
4.5 Explanation of Operations

4.1 Overview

The source clock (SRCCLK) is generated as the source of internal clocks used in operating this device.
This section explains generation and oscillation control of the source clock (SRCCLK) and selection of a clock as the source clock (SRCCLK).

■ Overview

This device operates with various internal clocks. The various internal clocks are generated by dividing the source clock (SRCCLK).

The following 3 clocks can be selected for the source clock (SRCCLK):

- Main clock (MCLK)
- PLL clock (PLLCLK)
- Sub clock (SBCLK)

The clock generating parts control the following:

- Main clock (MCLK) generation
  - Controls the oscillation of the main clock (MCLK).
  - Sets the oscillation stabilization wait time of the main clock (MCLK).
  - Controls the main timer or generation of main timer interrupt requests.
- Sub clock (SBCLK) generation
  - Controls the oscillation of the sub clock (SBCLK).
  - Sets the oscillation stabilization wait time of the sub clock (SBCLK).
  - Controls the sub timer or generation of sub timer interrupt requests.
- PLL clock (PLLCLK) generation
  - Controls the oscillation of the PLL clock (PLLCLK).
  - Sets the oscillation stabilization wait time of the PLL clock (PLLCLK).
  - Sets the PLL multiple rate (the main clock (MCLK) multiple rate for generating the PLL clock (PLLCLK)).
    The multiple rate can be set only for the main clock (MCLK), but not for the subclock (SBCLK).
- Source clock (SRCCLK) selection

  Selects one of 3 clocks for use as the source clock (SRCCLK).

4.2 Configuration

The clock generating parts consist of the clock generating parts themselves and the source clock (SRCCLK) selection block.

4.2.1 Clock Generating Parts

There are 3 clock generating parts. Any of the clocks generated by the clock generating parts can be selected for the source clock (SRCCLK).

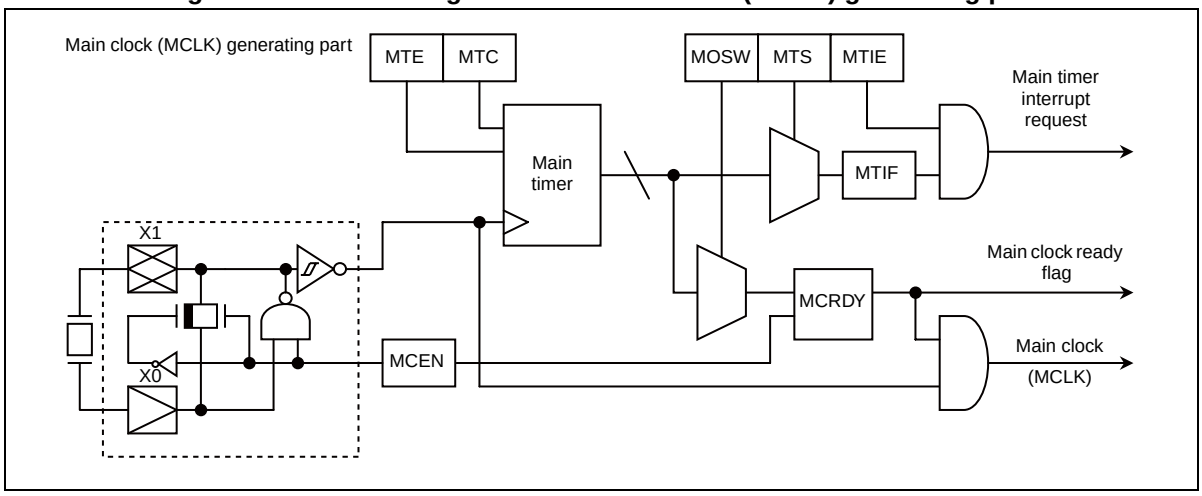
■ Main clock (MCLK) generating part

This part uses inputs from the X0 pin and X1 pin (main oscillator) to generate the main clock (MCLK).

The main clock (MCLK) is used to generate the PLL clock (PLLCLK).

Figure 4.2-1 shows a block diagram of the main clock (MCLK) generating part.

Figure 4.2-1 Block diagram of the main clock (MCLK) generating part



- Main timer

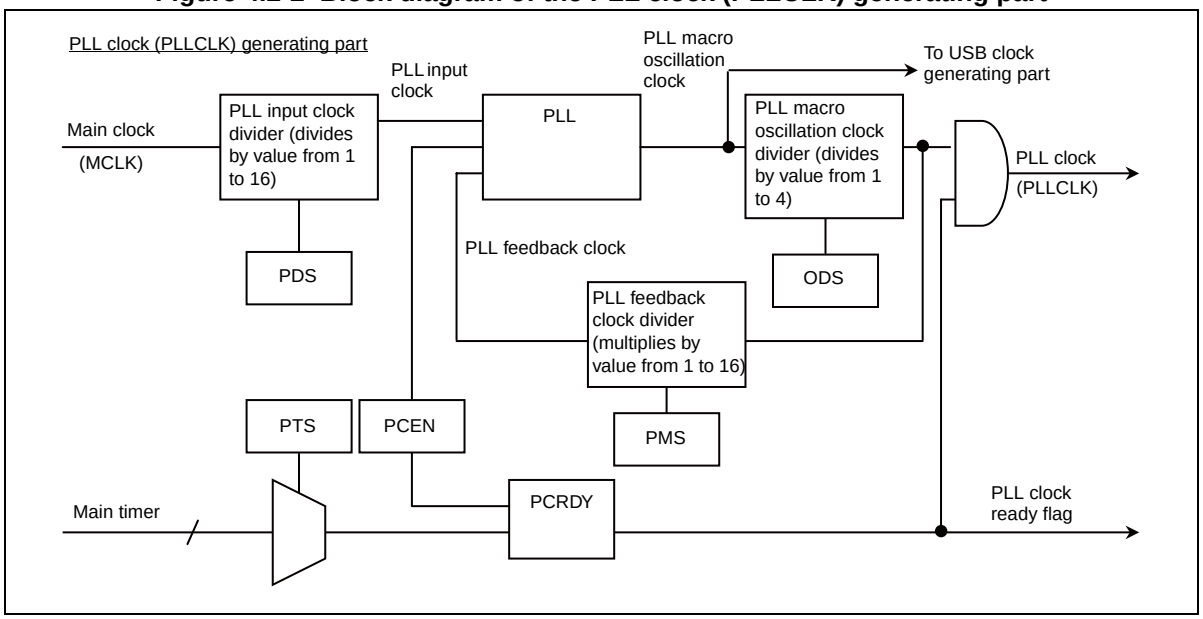
  The main timer operates with the main clock (MCLK). For details, see "CHAPTER 6 Main Timer".

■ PLL clock (PLLCLK) generating part

This part multiplies the main clock (MCLK) to generate the PLL clock (PLLCLK).

Figure 4.2-2 shows a block diagram of the PLL clock (PLLCLK) generating part.

Figure 4.2-2 Block diagram of the PLL clock (PLLCLK) generating part



- PLL

  Clock multiplication circuit
- PLL input clock divider

  This divider divides the main clock (MCLK) to generate the PLL input clock.
- PLL feedback clock divider

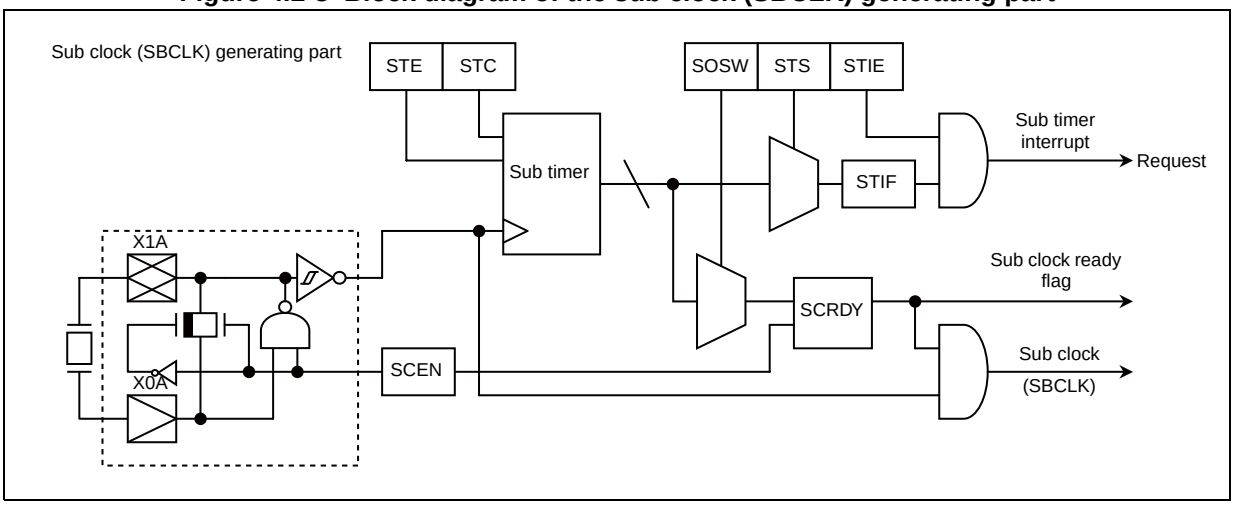
  This divider divides the PLL clock (PLLCLK) generated by dividing the PLL macro oscillation clock in order to generate the PLL feedback clock.
- PLL macro oscillation clock divider

  This divider divides the PLL macro oscillation clock to generate the PLL clock (PLLCLK).

■ Sub clock (SBCLK) generating part

This part uses inputs from the X0A pin and X1A pin (sub oscillator) to generate the sub clock (SBCLK).
The sub clock (SBCLK) is the oscillation output as is.
Figure 4.2-3 shows a block diagram of the sub clock (SBCLK) generating part.

Figure 4.2-3 Block diagram of the sub clock (SBCLK) generating part



- Sub timer
  The sub timer operates with the sub clock (SBCLK). For details, see "CHAPTER 7 Sub Timer".

4.2.2 Source Clock (SRCCLK) Selection Block

This section explains selection of the source clock (SRCCLK). The source clock (SRCCLK) is selected from the following 3 clock sources:

- Main clock (MCLK) divided by 2
- PLL clock (PLLCLK)
- Sub clock (SBCLK)

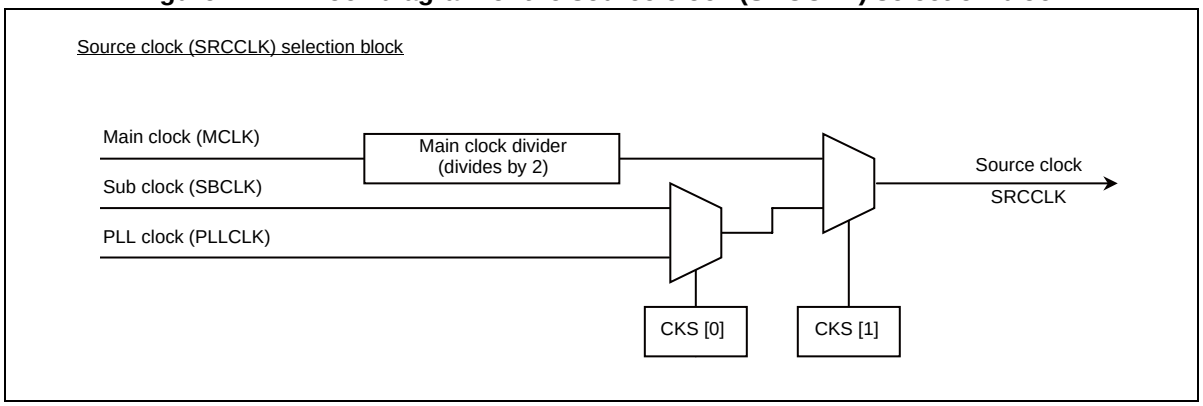
When an initialization reset (INIT) is generated, the settings of the source clock (SRCCLK) are initialized, and the main clock (MCLK) divided by 2 is set for the source clock (SRCCLK).

Change it to an arbitrary source clock (SRCCLK) with the setting of the clock source select register (CSELR) after the start of program operation.

■ Block diagram of the source clock (SRCCLK) selection block

Figure 4.2-4 shows a block diagram of the source clock (SRCCLK) selection block.

Figure 4.2-4 Block diagram of the source clock (SRCCLK) selection block



- Main clock divider (divides by 2)

  The divider divides the main clock (MCLK) by 2 and sets the resultant value for the source clock (SRCCLK).

- CKS1 and CKS0 bits

  These bits are the source clock (SRCCLK) selection bits in the clock source select register (CSELR).

4.3 Pins

This section explains the pins of the clock generating parts.

■ Overview

- X0 and X1 pins

  These pins are used to generate the main clock (MCLK).
- X0A and X1A pins

  These pins are used to generate the sub clock (SBCLK).

  They are used to connect the oscillator to an external unit.

  The pins are multiplexed pins. For details of using the X0A and X1A pins of the sub clock (SBCLK), see "2.4 Setting Method for Pins".

4.4 Registers

This section explains the configuration and functions of registers of the clock generating parts.

■ Registers of the clock generating parts

Table 4.4-1 lists the registers of the clock generating parts.

Table 4.4-1 Registers of the clock generating parts

Abbreviated Register Name	Register Name	Reference
CSELR	Clock source select register	4.4.1
CMONR	Clock source monitor register	4.4.2
CSTBR	Clock stabilization time select register	4.4.3
PLLCR	PLL configuration register	4.4.4

4.4.1 Clock Source Select Register (CSELR)

This register controls the clock source and selects the source clock (SRCCLK).

Figure 4.4-1 shows the bit configuration of the clock source select register (CSELR).

Figure 4.4-1 Bit configuration of the clock source select register (CSELR)

bit	7	6	5	4	3	2	1	0
	SCEN	PCEN	MCEN	Reserved	Reserved	Reserved	CKS1	CKS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	0	0	1	0	0	0	0	0
Initial value (at RST)	*	*	*	0	0	0	*	*

R/W: Read/Write

*: Uninitialized bit

<Notes>

- When this register is read, the actual setting value is not necessarily read. To verify that the value specified for this register has actually been made effective, read the clock source monitor register (CMONR).
- Before changing this register, verify that the value specified for this register is the same as the value of the clock source monitor register (CMONR).
- Writing of this register is ignored during switching of the clocks (CKS1, CKS0 ≠ CKM1, CKM0).

[bit7]: SCEN (Sub clock oscillation enable bit)

This bit controls the oscillation of the sub clock (SBCLK).

Written Value	Explanation	Remarks
0	The oscillation of the sub clock (SBCLK) is stopped.	The X0A or X1A pin can be used as a port (PK0, PK1).
1	The sub clock (SBCLK) starts oscillating.	The X0A and X1A pins are used to generate the sub clock (SBCLK).

<Notes>

- If the sub clock (SBCLK) is selected with the CKS1 and CKS0 bits (CKS1, CKS0=11) as the source clock (SRCCLK), this bit cannot be changed.
- The sub timer is cleared when "0" is written to the bit.
- In stop mode, the oscillation of the sub clock (SBCLK) is stopped regardless of the value of the bit.

[bit6]: PCEN (PLL clock oscillation enable bit)

This bit controls the oscillation of the PLL clock (PLLCLK).

Written Value	Explanation
0	The oscillation of the PLL clock (PLLCLK) is stopped.
1	The PLL clock (PLLCLK) starts oscillating.

<Notes>

- Write "0" to this bit to stop the oscillation of the PLL clock (PLLCLK) before entering stop mode.
- The bit cannot be changed under any of the following conditions:
  - When the PLL clock (PLLCLK) is selected with the CKS1 and CKS0 bits (CKS1, CKS0 = 10) as the source clock (SRCCLK)
  - When the oscillation of the main clock (MCLK) is stopped, or the oscillation stabilization wait time is in effect
    (MCRDY bit = 0 in the clock source monitor register (CMONR))
- This bit is changed to "0" when the MCEN bit (MCEN = 0) is specified to stop the oscillation of the main clock (MCLK).
- Do not change this bit from "0" to "1" while the main timer is being cleared (MTC bit = 1 in the main timer control register (MTMCR)).
- If this bit is changed from "0" to "1" to enable the oscillation of the PLL clock (PLLCLK), the main timer is cleared.

  In such cases, "1" is read from the MTC bit in the main timer control register (MTMCR).

[bit5]: MCEN (Main clock oscillation enable bit)

This bit controls the oscillation of the main clock (MCLK).

Written Value	Explanation
0	The oscillation of the main clock (MCLK) is stopped.
1	The main clock (MCLK) starts oscillating.

<Notes>

- If any of the following is selected with the CKS1 or CKS0 bit as the source clock (SRCCLK), this bit cannot be changed.
  - The main clock (MCLK) is selected (CKS1, CKS0 = 00 or 01).
  - The PLL clock (PLLCLK) is selected (CKS1, CKS0 = 10).
- The main timer is cleared when "0" is written to this bit.
- In stop mode, the oscillation of the main clock (MCLK) is stopped regardless of the value of the bit.

[bit4 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: CKS1, CKS0 (Source clock select bits)

These bits select the source clock (SRCCLK).

CKS1	CKS0	Explanation
0	0	Main clock (MCLK) divided by 2
0	1	
1	0	PLL clock (PLLCLK)
1	1	Sub clock (SBCLK)

A clock whose oscillation is stopped or that has entered the oscillation stabilization wait time cannot be selected as the source clock (SRCCLK).

Furthermore, no switching from the PLL clock (PLLCLK) to the sub clock (SBCLK) or from the sub clock (SBCLK) to the PLL clock (PLLCLK) is possible.

Table 4.4-2 lists the conditions for changes of this bit.

Table 4.4-2 CKS1 and CKS0 bit change conditions

Value before Change		Changeable Value [CKS1:CKS0]	Change Condition Bit Clock Source Monitor Register (CMONR)	Unchangeable Value [CKS1:CKS0]
CKS1	CKS0			
0	0	00, 01	MCRDY = 1	11
		10	PCRDY = 1	
0	1	00, 01	MCRDY = 1	10
		11	SCRDY = 1	
1	0	00	MCRDY = 1	01, 11
		10	PCRDY = 1	
1	1	01	MCRDY = 1	00, 10
		11	SCRDY = 1	

Do not write the unchangeable values listed in Table 4.4-2. For the procedures for switching the source clock (SRCCLK), see "4.5.2 Switching the Source Clock (SRCCLK)".

4.4.2 Clock Source Monitor Register (CMONR)

This register displays the clock source and state of the source clock (SRCCLK).

The value specified for the clock source select register (CSELR) can be verified by reading this register to verify whether it is actually effective.

Figure 4.4-2 shows the bit configuration of the clock source monitor register (CMONR).

Figure 4.4-2 Bit configuration of the clock source monitor register (CMONR)

bit	7	6	5	4	3	2	1	0
	SCRDY	PCRDY	MCRDY	Reserved	Reserved	Reserved	CKM1	CKM0
Attribute	R	R	R	R	R	R	R	R
Initial value (at INIT)	0	0	1	0	0	0	0	0
Initial value (at RST)	*	*	*	0	0	0	*	*

R: Read only

*: Uninitialized bit

<Notes>

- When changing a set value of the clock source select register (CSELR), be sure to read this register and verify that the read value is the same as the set value of the clock source select register (CSELR).
- Do not change the clock source select register (CSELR) unless the set value of the clock source select register (CSELR) matches the register value.

[bit7]: SCRDY (Sub clock ready bit)

This bit displays the sub clock (SBCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

<Notes>

- If this bit is "0", the sub clock (SBCLK) cannot be selected as the source clock (SRCCLK).
- After the SCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit6]: PCRDY (PLL clock ready bit)

This bit displays the PLL clock (PLLCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

<Notes>

- If this bit is "0", the PLL clock (PLLCLK) cannot be selected as the source clock (SRCCLK).
- After the PCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit5]: MCRDY (Main clock ready bit)

This bit displays the main clock (MCLK) state.

Read Value	Explanation
0	The oscillation is stopped, or the oscillation stabilization wait time is in effect.
1	The oscillation stabilization is in effect. This clock can be used as the source clock (SRCCLK).

<Notes>

- If this bit is "0", neither the main clock (MCLK) nor the PLL clock (PLLCLK) can be selected as the source clock (SRCCLK).
- After the MCEN bit in the clock source select register (CSELR) is changed from "1" to "0", this bit may be read as having a value of "1".

[bit4 to bit2]: Reserved bits

In case of reading	"0" is read.

[bit1, bit0]: CKM1, CKM0 (Source clock display bits)

These bits display the clock selected as the source clock (SRCCLK).

CKM1	CKM0	Explanation
0	0	The main clock (MCLK) divided by 2 is selected.
0	1	
1	0	The PLL clock (PLLCLK) is selected.
1	1	The sub clock (SBCLK) is selected.

4.4.3 Clock Stabilization Time Select Register (CSTBR)

This register sets the oscillation stabilization wait time of the clock source.

The oscillation stabilization wait time set in this register is used under the following conditions with the ready bit being "1" for the relevant clock:

- When returning from stop mode or watch mode
- When the main oscillation is stopped and an initialize reset (INIT) is generated
- When clock oscillation is enabled after being stopped

  The ready bits are as follows:
  - Sub clock: SCRDY bit
  - PLL clock: PCRDY bit
  - Main clock: MCRDY bit

Figure 4.4-3 shows the bit configuration of the clock stabilization select register (CSTBR).

Figure 4.4-3 Bit configuration of the clock stabilization time select register (CSTBR)

bit	7	6	5	4	3	2	1	0
	Reserved	SOSW2	SOSW1	SOSW0	MOSW3	MOSW2	MOSW1	MOSW0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
$\overline{\text{INIT}}$ pin = "L" level	0	0	0	0	0	0	0	0
Initial value (at INIT)	0	0	0	0	*	*	*	*
Initial value (at RST)	0	*	*	*	*	*	*	*

R/W: Read/Write

*: Uninitialized bit

<Note>

When the main oscillation is stopped and an initialize reset (INIT) is generated the main oscillation stabilization wait time after operation is restarted is the initial value of this register.

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6 to bit4]: SOSW2 to SOSW0 (Sub clock oscillation stabilization wait select bits)

These bits select the oscillation stabilization wait time of the sub clock (SBCLK).

SOSW2	SOSW1	SOSW0	Sub Clock (SBCLK) Oscillation Stabilization Wait Time	At 32.768 kHz
0	0	0	$2^8 \times$ Sub clock (SBCLK) period	About 7.8 ms
0	0	1	$2^9 \times$ Sub clock (SBCLK) period	About 15.6 ms
0	1	0	$2^{10} \times$ Sub clock (SBCLK) period	About 31.3 ms
0	1	1	$2^{11} \times$ Sub clock (SBCLK) period	62.5 ms
1	0	0	$2^{12} \times$ Sub clock (SBCLK) period	125.0 ms
1	0	1	$2^{13} \times$ Sub clock (SBCLK) period	250.0 ms
1	1	0	$2^{14} \times$ Sub clock (SBCLK) period	500.0 ms
1	1	1	$2^{15} \times$ Sub clock (SBCLK) period	1 s

<Notes>

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Writing to this bit is ignored when the following conditions are satisfied (in the oscillation stabilization wait time of the sub clock (SBCLK)):
  - SCRDY bit = 0 in the clock source monitor register (CMONR)
  - SCEN bit = 1 in the clock source select register (CSELR)

[bit3 to bit0]: MOSW3 to MOSW0 (Main clock oscillation stabilization select bits)

These bits select the oscillation stabilization wait time of the main clock (MCLK).

MOSW3	MOSW2	MOSW1	MOSW0	Main Clock (MCLK) Oscillation Stabilization Wait Time	At 4 MHz	At 8 MHz	At 48 MHz
0	0	0	0	$2^1 \times$ Main clock (MCLK) period	500 ns	250 ns	About 42 ns
0	0	0	1	$2^5 \times$ Main clock (MCLK) period	8 µs	4 µs	About 667 ns
0	0	1	0	$2^6 \times$ Main clock (MCLK) period	16 µs	8 µs	About 1 µs
0	0	1	1	$2^7 \times$ Main clock (MCLK) period	32 µs	16 µs	About 3 µs
0	1	0	1	$2^8 \times$ Main clock (MCLK) period	64 µs	32 µs	About 5 µs
0	1	0	1	$2^9 \times$ Main clock (MCLK) period	128 µs	64 µs	About 11 µs
0	1	1	0	$2^{10} \times$ Main clock (MCLK) period	256 µs	128 µs	About 21 µs
0	1	1	1	$2^{11} \times$ Main clock (MCLK) period	512 µs	256 µs	About 43 µs
1	0	0	0	$2^{12} \times$ Main clock (MCLK) period	About 1 ms	512 µs	About 85 µs
1	0	0	1	$2^{13} \times$ Main clock (MCLK) period	About 2 ms	About 1 ms	About 171 µs
1	0	1	0	$2^{14} \times$ Main clock (MCLK) period	About 4 ms	About 2 ms	About 341 µs
1	0	1	1	$2^{15} \times$ Main clock (MCLK) period	About 8 ms	About 4 ms	About 683 µs
1	1	0	0	$2^{17} \times$ Main clock (MCLK) period	About 33 ms	About 16 ms	About 3 ms
1	1	0	1	$2^{19} \times$ Main clock (MCLK) period	About 131 ms	About 66 ms	About 11 ms
1	1	1	0	$2^{21} \times$ Main clock (MCLK) period	About 524 ms	About 262 ms	About 44 ms
1	1	1	1	$2^{23} \times$ Main clock (MCLK) period	About 2 s	About 1 s	About 175 ms

<Notes>

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Specify an oscillation stabilization wait time as 25µs or longer for a product equipped with a regulator.
- Writing to this bit is ignored when the following conditions are satisfied (in the oscillation stabilization wait time of the main clock (MCLK)):
  - MCRDY bit = 0 in the clock source monitor register (CMONR)
  - MCEN bit = 1 in the clock source select register (CSELR)

4.4.4 PLL Configuration Register (PLLCR)

This register sets the multiple rate for generating the PLL clock (PLLCLK) from the main clock (MCLK).

For the calculation of the clock frequency and the multiple rate related to generating the PLL clock (PLLCLK), see "4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)".

Figure 4.4-4 shows the bit configuration of the PLL configuration register (PLLCR).

Figure 4.4-4 Bit configuration of the PLL configuration register (PLLCR)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	ODS1	ODS0	PMS3	PMS2	PMS1	PMS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	0	0	0	0	0	0	0	0
Initial value (at RST)	0	0	*	*	*	*	*	*

bit	7	6	5	4	3	2	1	0
	PTS3	PTS2	PTS1	PTS0	PDS3	PDS2	PDS1	PDS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (at INIT)	1	1	1	1	0	0	0	0
Initial value (at RST)	*	*	*	*	*	*	*	*

R/W: Read/Write

*: Uninitialized bit

<Note>

Writing to this bit is ignored when the oscillation of the PLL clock (PLLCLK) is enabled (PCEN = 1 in the clock source select register (CSELR)).

[bit15, bit14]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: ODS1, ODS0 (PLL macro oscillation clock division rate select bits)

These bits select the division rate from the PLL macro oscillation clock to the PLL clock (PLLCLK).

ODS1	ODS0	Explanation
0	0	PLL clock (PLLCLK) = PLL macro oscillation clock / 1
0	1	PLL clock (PLLCLK) = PLL macro oscillation clock / 2
1	0	PLL clock (PLLCLK) = PLL macro oscillation clock / 3
1	1	PLL clock (PLLCLK) = PLL macro oscillation clock / 4

[bit11 to bit8]: PMS3 to PMS0 (PLL clock multiple rate select bits)

These bits select the multiple rate from the PLL input clock to the PLL clock (PLLCLK).

PMS3	PMS2	PMS1	PMS0	PLL Clock (PLLCLK) Multiple Rate
0	0	0	0	PLL clock (PLLCLK) = PLL input clock × 1
0	0	0	1	PLL clock (PLLCLK) = PLL input clock × 2
0	0	1	0	PLL clock (PLLCLK) = PLL input clock × 3
0	0	1	1	PLL clock (PLLCLK) = PLL input clock × 4
0	1	0	0	PLL clock (PLLCLK) = PLL input clock × 5
0	1	0	1	PLL clock (PLLCLK) = PLL input clock × 6
0	1	1	0	PLL clock (PLLCLK) = PLL input clock × 7
0	1	1	1	PLL clock (PLLCLK) = PLL input clock × 8
1	0	0	0	PLL clock (PLLCLK) = PLL input clock × 9
1	0	0	1	PLL clock (PLLCLK) = PLL input clock × 10
1	0	1	0	PLL clock (PLLCLK) = PLL input clock × 11
1	0	1	1	PLL clock (PLLCLK) = PLL input clock × 12
1	1	0	0	PLL clock (PLLCLK) = PLL input clock × 13
1	1	0	1	PLL clock (PLLCLK) = PLL input clock × 14
1	1	1	0	PLL clock (PLLCLK) = PLL input clock × 15
1	1	1	1	PLL clock (PLLCLK) = PLL input clock × 16

[bit7 to bit4]: PTS3 to PTS0 (PLL clock oscillation stabilization wait time select bits)

These bits select the oscillation stabilization wait time of the PLL clock (PLLCLK).

PTS3	PTS2	PTS1	PTS0	PLL Clock (PLLCLK) Oscillation Stabilization Wait Time	At 4 MHz	At 8 MHz	At 48 MHz
1	0	0	0	$2^9 \times$ Main clock (MCLK) period	128.0 µs	64.0 µs	About 10.7 µs
1	0	0	1	$2^{10} \times$ Main clock (MCLK) period	256.0 µs	128.0 µs	About 21.3 µs
1	0	1	0	$2^{11} \times$ Main clock (MCLK) period	512.0 µs	256.0 µs	About 42.7 µs
1	0	1	1	$2^{12} \times$ Main clock (MCLK) period	About 1 ms	512.0 µs	About 85.3 µs
1	1	0	0	$2^{13} \times$ Main clock (MCLK) period	About 2 ms	About 1 ms	About 170.7 µs
1	1	0	1	$2^{14} \times$ Main clock (MCLK) period	About 4 ms	About 2 ms	About 341.3 µs
1	1	1	0	$2^{15} \times$ Main clock (MCLK) period	About 8 ms	About 4 ms	About 682.7 µs
1	1	1	1	$2^{16} \times$ Main clock (MCLK) period	About 16.4 ms	About 8 ms	About 1.4 ms

<Notes>

- The times listed in the table are calculated values. Use these values only as a guide because the actual times may include some errors depending on the oscillation state.
- Always write "1" to the PTS3 bit.

[bit3 to bit0]: PDS3 to PDS0 (PLL input clock division select bits)

These bits select the main clock (MCLK) division rate for generating the PLL input clock.

PDS3	PDS2	PDS1	PDS0	PLL Input Clock Division Selection
0	0	0	0	PLL input clock = Main clock (MCLK) / 1
0	0	0	1	PLL input clock = Main clock (MCLK) / 2
0	0	1	0	PLL input clock = Main clock (MCLK) / 3
0	0	1	1	PLL input clock = Main clock (MCLK) / 4
0	1	0	0	PLL input clock = Main clock (MCLK) / 5
0	1	0	1	PLL input clock = Main clock (MCLK) / 6
0	1	1	0	PLL input clock = Main clock (MCLK) / 7
0	1	1	1	PLL input clock = Main clock (MCLK) / 8
1	0	0	0	PLL input clock = Main clock (MCLK) / 9
1	0	0	1	PLL input clock = Main clock (MCLK) / 10
1	0	1	0	PLL input clock = Main clock (MCLK) / 11
1	0	1	1	PLL input clock = Main clock (MCLK) / 12
1	1	0	0	PLL input clock = Main clock (MCLK) / 13
1	1	0	1	PLL input clock = Main clock (MCLK) / 14
1	1	1	0	PLL input clock = Main clock (MCLK) / 15
1	1	1	1	PLL input clock = Main clock (MCLK) / 16

4.5 Explanation of Operations

This section explains the operations of the clock generating parts.

This section explains the operations of each clock source and how the source clocks are switched.

4.5.1 Explanation of Clock Source Operations

This section explains mainly oscillation control of the clock sources.

■ Main clock (MCLK)

This clock is generated with inputs from the X0 pin and X1 pin (main oscillator). It is used to generate the PLL clock.

The main clock is used in operating the main timer. (See "CHAPTER 6 Main Timer".)

● Conditions for stopping oscillation

The oscillation of the main clock (MCLK) stops under any of the following conditions:

- When stop mode is in effect
- When the sub clock (SBCLK) is selected for the source clock (SRCCLK) and the oscillation of the main clock (MCLK) is stopped (that is, when the following conditions are satisfied):
  - CKS1 or CKS0 bit in the clock source select register (CSELR) = 11
  - MCEN bit in the clock source select register (CSELR) = 0

Supplying of the main clock (MCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the MOSW3 to MOSW0 bits in the clock stabilization time select register (CSTBR) has elapsed.

● Selecting the oscillation stabilization wait time

Supplying of the main clock (MCLK) starts after a wait for the oscillation of the main clock to stabilize once the oscillation has been enabled.

The MOSW3 to MOSW0 bits in the clock stabilization time select register (CSTBR) specify the oscillation stabilization wait time of the main clock (MCLK).

Input at the "L" level to the $\overline{\text{INIT}}$ pin initializes the MOSW3 to MOSW0 bits, returning the oscillation stabilization wait time to its initial value. In such cases, the initial value is $2^1 \times$ Main clock (MCLK) period.

The MOSW3 to MOSW0 bits are not initialized by any other reset that occurs.

<Note>

Specify an oscillation stabilization wait time as 25µs or longer for products equipped with regulators.

● End of the oscillation stabilization wait time

The main clock (MCLK) is supplied at the end of the oscillation stabilization wait time.

Checking the following values enables you to verify whether the main clock (MCLK) has entered the oscillation stabilization wait time while operation of the main clock (MCLK) is enabled.

Oscillation Stabilization Wait State Display	Oscillation Stabilization State Display
MCRDY = 0 in the clock source monitor register (CMONR)	MCRDY = 1 in the clock source monitor register (CMONR)

■ PLL clock (PLLCLK)

This high-performance clock multiplies and generates the main clock (MCLK).

● Conditions for stopping oscillation

The oscillation of the PLL clock (PLLCLK) stops under any of the following conditions:

- When the oscillation of the main clock (MCLK) is stopped, or the oscillation stabilization wait time is in effect

  (PCEN bit = 0 in the clock source select register (CSELR))
- When the following conditions are satisfied and a clock other than the PLL clock (PLLCLK) is selected for the source clock (SRCCLK):
  - CKS1 or CKS0 bit in the clock source select register (CSELR) = a value other than 10
  - PCEN bit in the clock source select register (CSELR) = 0

Supplying of the PLL clock (PLLCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the PTS3 to PTS0 bits in the PLL configuration register (PLLCR) has elapsed.

Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the PCEN bit in the clock source select register (CSELR) to "0" and stops the oscillation of the PLL clock (PLLCLK). (To start the oscillation after such initialization, set the PCEN bit in the clock source select register (CSELR) to "1".)

● Selecting an oscillation stabilization wait time

Supplying of the PLL clock (PLLCLK) starts after a wait for the oscillation of the PLL clock to stabilize once the oscillation has been enabled.

The PTS3 to PTS0 bits in the PLL configuration register (PLLCR) specify the oscillation stabilization wait time of the PLL clock (PLLCLK).

Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the PTS3 to PTS0 bits, returning the oscillation stabilization wait time to its initial value. In such cases, the initial value is $2^{16} \times$ Main clock (MCLK) period.

To change the oscillation stabilization wait time, set the PTS3 to PTS0 bits, and then write "1" to the PCEN bit in the clock source select register (CSELR).

● End of the oscillation stabilization wait time

The PLL clock (PLLCLK) is supplied at the end of the oscillation stabilization wait time.

Checking the following values enables you to verify whether the PLL clock (PLLCLK) has entered the oscillation stabilization wait time while operation of the PLL clock (PLLCLK) is enabled.

Oscillation stabilization wait state display	Oscillation stabilization state display
PCRDY = 0 in the clock source monitor register (CMONR)	PCRDY = 1 in the clock source monitor register (CMONR)

■ Sub clock (SBCLK)

This clock is generated with inputs from the X0A pin and X1A pin (sub oscillator). The sub clock (SBCLK) is the oscillation output as is.

The sub clock is used in operating the sub timer (See "CHAPTER 7 Sub Timer").

● Conditions for stopping oscillation

The oscillation of the sub clock (SBCLK) stops under any of the following conditions:

- When input to the $\overline{\text{INIT}}$ pin is at the "L" level
- When stop mode is in effect
- When any clock other than the sub clock (SBCLK) is selected for the source clock (SRCCLK) and the oscillation of the sub clock (SBCLK) is stopped (that is, when the following conditions are satisfied):
  - CKS1 or CKS0 bit in the clock source select register (CSELR) = a value other than 11
  - SCEN bit in the clock source select register (CSELR) = 0
- The pins are set to use ports (the pins are multiplexed for the sub clock (SBCLK) generating part and the ports).

Supplying of the sub clock (SBCLK) starts after all the above oscillation stop conditions are cleared and the oscillation stabilization wait time specified by the SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR) has elapsed.

Input at the "L" level to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the SCEN bit in the clock source select register (CSELR) to "0" and stops the oscillation of the sub clock (SBCLK). (To start the oscillation after such initialization, set the SCEN bit in the clock source select register (CSELR) to "1".)

● Selecting an oscillation stabilization wait time

Supplying of the sub clock (SBCLK) starts after a wait for the oscillation of the sub clock to stabilize once the oscillation has been enabled.

The SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR) specify the oscillation stabilization wait time of the sub clock (SBCLK).

Input at the "L" to the $\overline{\text{INIT}}$ pin or a return from an initialization reset (INIT) initializes the SOSW2 to SOSW0 bits, returning the oscillation wait time to its initial value. In such cases, the initial value is $2^8 \times$ Sub clock (SBCLK) period.

To change the oscillation stabilization wait time, set the SOSW2 to SOSW0 bits.

● End of the oscillation stabilization wait time

The sub clock (SBCLK) is supplied at the end of the oscillation stabilization wait time.

Checking the following values enables you to verify whether the sub clock (SBCLK) has entered the oscillation stabilization wait time while operation of the sub clock (SBCLK) is enabled.

Oscillation stabilization wait state display	Oscillation stabilization state display
SCRDY = 0 in the clock source monitor register (CMONR)	SCRDY = 1 in the clock source monitor register (CMONR)

4.5.2 Switching the Source Clock (SRCCLK)

This section explains switching of the source clock (SRCCLK).

■ Overview

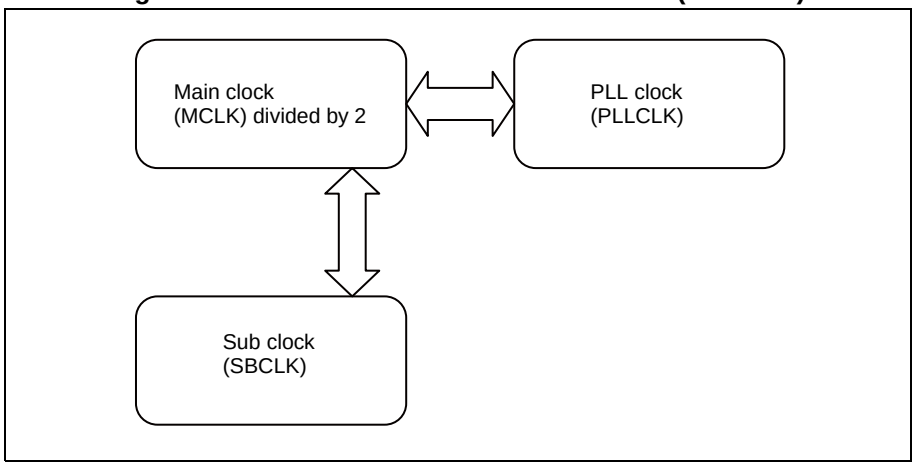
When "L" is input to the $\overline{\text{INIT}}$ pin or an initialization reset (INIT) is generated, the settings of the source clock (SRCCLK) are initialized, and the main clock (MCLK) divided by 2 is set for the source clock (SRCCLK).

The CKS1 and CKS0 bits of the clock source select register (CSELR) can be used to select the source clock (SRCCLK) from the clock sources after the start of program operation.

For this change to the source clock (SRCCLK), no switch from the PLL clock (PLLCLK) to the sub clock (SBCLK) or from the sub clock (SBCLK) to the PLL clock (PLLCLK) is possible. To do so, specify the main clock (MCLK) divided by 2, and then switch it.

Figure 4.5-1 shows how to switch the source clock (SRCCLK).

Figure 4.5-1 How to switch the source clock (SRCCLK)



<Note>

Even if the source clock (SRCCLK) is switched, the oscillation enable settings (the values of the SCEN bit, PCEN bit, and MCEN bit in the clock source select register (CSELR)) of each clock are maintained. Stop the oscillation as necessary.

■ Procedures

● Switching from the main clock (MCLK) divided by 2 to the PLL clock (PLLCLK)

To switch the source clock (SRCCLK) from the main clock (MCLK) divided by 2 to the PLL clock (PLLCLK), make settings by following the procedure below.

1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 00 or 01) of the clock source monitor register (CMONR) to verify that the main clock (MCLK) divided by 2 is selected.
2. Set the PLL multiple rate and the PLL clock (PLLCLK) oscillation stabilization wait time in the PLL configuration register (PLLCR).
3. Set the PCEN bit (PCEN=1) in the clock source select register (CSELR) to start the oscillation of the PLL clock (PLLCLK).
4. Check the PCRDY bit (PCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the PLL clock (PLLCLK) has stabilized.
5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 10) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the PLL clock (PLLCLK).
6. Check the CKM1 and CKM0 bits (CKM1, CKM = 10) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the PLL clock (PLLCLK).

<Note>

If the oscillation of the PLL clock (PLLCLK) has been enabled, steps 2 to 4 can be omitted.

● Switching from the PLL clock (PLLCLK) to the main clock (MCLK) divided by 2

To switch the source clock (SRCCLK) from the PLL clock (PLLCLK) to the main clock (MCLK) divided by 2, make settings by following the procedure below.

1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 10) in the clock source monitor register (CMONR) to verify that the PLL clock (PLLCLK) is selected.
2. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 00) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the main clock (MCLK) divided by 2.
3. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 00) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the main clock (MCLK) divided by 2.

● Switching from the main clock (MCLK) divided by 2 to the sub clock (SBCLK)

To switch the source clock (SRCCLK) from the main clock (MCLK) divided by 2 to the sub clock (SBCLK), make settings by following the procedure below.

1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 01) in the clock source monitor register (CMONR) to verify that the main clock (MCLK) divided by 2 is selected.
2. Set the oscillation stabilization wait time of the sub clock (SBCLK) in the SOSW2 to SOSW0 bits in the clock stabilization time select register (CSTBR).
3. Set the SCEN bit (SCEN=1) in the clock source select register (CSELR) to start the oscillation of the sub clock (SBCLK).
4. Check the SCRDY bit (SCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the sub clock (SBCLK) has stabilized.
5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 11) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the sub clock (SBCLK).
6. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 11) in the clock source monitor register (CMONR) to verify that the source clock (SRCCLK) was switched to the sub clock (SBCLK).

<Note>

If the oscillation of the sub clock (SBCLK) has been enabled, steps 2 to 4 can be omitted.

● Switching from the sub clock (SBCLK) to the main clock (MCLK) divided by 2

To switch the source clock (SRCCLK) from the sub clock (SBCLK) to the main clock (MCLK) divided by 2, make settings by following the procedure below.

1. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 11) in the clock source monitor register (CMONR) to verify that the sub clock (SBCLK) is selected.
2. Set the oscillation stabilization wait time of the main clock (MCLK) in the MOSW2 to MOSW0 bits in the clock stabilization time select register (CSTBR).
3. Set the MCEN bit (MCEN=1) in the clock source select register (CSELR) to start the oscillation of the main clock (MCLK).
4. Check the MCRDY bit (MCRDY = 1) in the clock source monitor register (CMONR) to verify that the oscillation of the main clock (MCLK) has stabilized.
5. Set the CKS1 and CKS0 bits (CKS1, CKS0 = 01) in the clock source select register (CSELR) to switch the source clock (SRCCLK) to the main clock (MCLK).
6. Check the CKM1 and CKM0 bits (CKM1, CKM0 = 01) in the clock source monitor register (CMONR) to verify that the source clock (CRCCLK) was switched to the main clock (MCLK).

<Note>

If the oscillation of the main clock (MCLK) has been enabled, steps 2 to 4 can be omitted.

4.5.3 Multiple Rate for Generating the PLL Clock (PLLCLK)

This section explains how to calculate the clock frequency and the multiple rate related to generating the PLL clock (PLLCLK).

PLL input clock frequency

= (Main oscillation frequency)/(Division rate set in the PDS bit in the PLL configuration register (PLLCR))

PLL multiple rate

= (Division rate set in the ODS bit in the PLL configuration register (PLLCR)) × (Multiple rate set in the PMS bit in the PLL configuration register (PLLCR))

PLL macro oscillation clock frequency

= (PLL input clock frequency) × PLL multiple rate

PLL clock (PLLCLK) frequency

= (PLL input clock frequency) × (Multiple rate set in the PMS bit in the PLL configuration register (PLLCR))

Table 4.5-1 lists sample settings of the PLL clock (PLLCLK).

Table 4.5-1 Sample settings of the PLL clock (PLLCLK)

Main Oscillation Frequency	PLL Configuration Register (PLLCR)			PLL Input Clock Frequency	PLL Multiple Rate ODS × PMS	PLL Macro Oscillation Clock Frequency	PLL Clock Frequency
	PDS3 to PDS0	ODS1 to ODS0	PMS3 to PMS0				
4 MHz	0000	00	0111	4 MHz	Multiplied by 8	32 MHz	32 MHz
4.167 MHz	0000	00	0111	4.167 MHz	Multiplied by 8	33 MHz	33 MHz
48MHz	0010	10	0001	16 MHz	Multiplied by 6	96 MHz	32 MHz
4MHz	0000	10	0111	4 MHz	Multiplied by 24	96 MHz	32 MHz
8MHz	0000	10	0011	8 MHz	Multiplied by 12	96 MHz	32 MHz

<Notes>

- The following conditions must be satisfied by the specified PLL input clock, PLL multiple rate, and PLL macro oscillation clock.

PLL Input Clock Frequency	**4 to 24 MHz**
PLL Multiple Rate	**Multiplied by 2 to 24**
PLL Macro Oscillation Clock Frequency	**48 to 96 MHz**

- It is prohibited to set ODS=00 and PMS=0000 (PLL multiply rate=1).

CHAPTER 5 Clock Division Control Part

This chapter explains the clock division control part that generates internal clocks.

5.1 Overview
5.2 Internal Clocks
5.3 Configuration
5.4 Registers
5.5 Division Rate

5.1 Overview

Internal clocks are generated by dividing the source clock (SRCCLK) input from a clock generating part.

The clock division control part divides the source clock (SRCCLK) and generates internal clocks to supply them to the CPU, bus, and/or peripheral functions.

Table 5.1-1 lists the internal clocks that are generated. These clocks are collectively called internal clocks.

Table 5.1-1 Internal clocks that are generated

Clock Name	Generation Source Clock
Base clock (BCLK)	Source clock (SRCCLK) divided by a value from 1 to 8
CPU clock (CCLK)	Base clock (BCLK) divided by 1 (undivided)
On-chip bus clock (HCLK)	Base clock (BCLK) divided by 1 (undivided)
External bus clock (TCLK)	Base clock (BCLK) divided by a value from 1 to 8
Peripheral clock (PCLK)	Base clock (BCLK) divided by a value from 1 to 16

For details of the source clock (SRCCLK), see "CHAPTER 4 Clock Generating Parts".

5.2 Internal Clocks

This section explains the internal clocks.

■ Base clock (BCLK)

This clock is the generation source of all internal clocks.

The DIVB2 to DIVB0 bits of the divide clock configuration register 0 (DIVR0) are used when this clock is generated by dividing the source clock (SRCCLK) by a value ranging from 1 to 8.

The clock can decrease at once the operating frequency of the entire device.

It is stopped in one of the following low-power dissipation modes:

- Watch mode
- Stop mode

■ CPU clock (CCLK)

This clock is supplied to the CPU in this device and generated from the base clock (BCLK).

Since it is generated without dividing the base clock (BCLK), the operating frequency is always the same as that for the base clock (BCLK).

It is stopped in one of the following low-power dissipation modes:

- Doze mode (during a stop time)
- Sleep mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
CPU clock (CCLK)	CPU (instruction execution block)

■ On-chip bus clock (HCLK)

This clock is supplied to the on-chip bus and each circuit connected to the on-chip bus. It is generated from the base clock (BCLK).

Since it is generated without dividing the base clock (BCLK), the operating frequency is always the same as that for the base clock (BCLK).

It is stopped in one of the following low-power dissipation modes:

- Bus sleep mode
- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
On-chip bus clock (HCLK)	DMA controller (DMAC)

■ External bus clock (TCLK)

This clock is supplied to an external bus interface.

The DIVT2 to DIVT0 bits of divide clock configuration register 1 (DIVR1) are used when this clock is generated by dividing the base clock (BCLK) by a value ranging from 1 to 8.

If there is no on-chip bus access in bus sleep mode, the clock can be stopped by specifying the TSTP bit in divide clock configuration register 1 (DIVR1).

It is stopped in one of the following low-power dissipation modes regardless of the setting:

- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
External bus clock (TCLK)	External bus interface

<Notes>

- The same frequency as that for the external bus clock (TCLK) is output for the bus clock (SYSCLK) from the SYSCLK pin.
- If an odd number is specified for the division rate of the base clock (BCLK) (DIVT2 to DIVT0 bits in divide clock configuration register 1 (DIVR1)), the duty ratio of the bus clock (SYSCLK) output from the SYSCLK pin cannot be 50%. The "H" level output period becomes 50% or less of the output period.

■ Peripheral clock (PCLK)

This clock is supplied to the peripheral buses and each peripheral function connected to the buses.

The DIVP3 to DIVP0 bits of divide clock configuration register 2 (DIVR2) are used when this clock is generated by dividing the base clock (BCLK) by a value ranging from 1 to 16.

It is stopped in one of the following low-power dissipation modes regardless of the setting:.

- Watch mode
- Stop mode

Clock Name	Typical Supply Destination
Peripheral clock (PCLK)	Peripheral bus Clock control part Reset controller Watchdog timer Interrupt controller External interrupt Delay interrupt 16-bit reload timer Each peripheral function

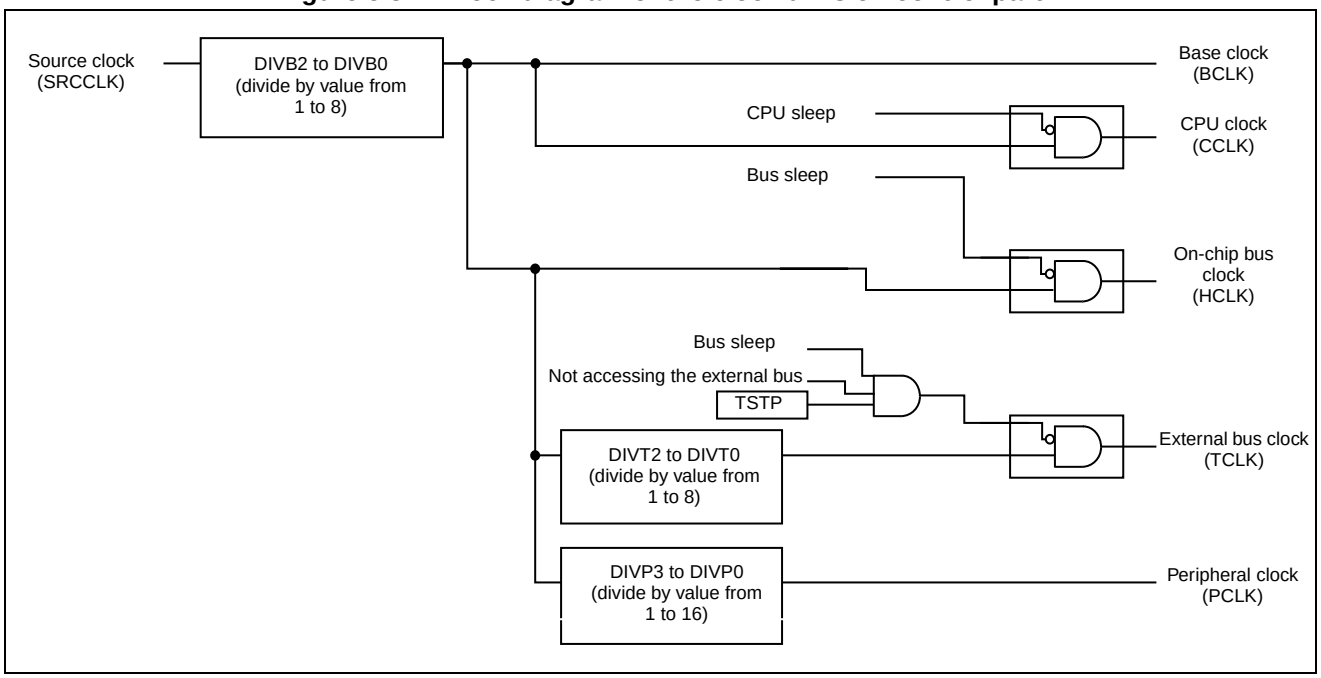
5.3 Configuration

The source clock input from a clock generating part is divided by the value specified in a register and output to a circuit.

■ Block diagram of the clock division control part

Figure 5.3-1 is a block diagram of the clock division control part.

Figure 5.3-1 Block diagram of the clock division control part



5.4 Registers

This section explains the configuration and functions of registers of the clock division control part.

■ Registers of the clock division control part

Table 5.4-1 lists the registers of the clock division control part.

Table 5.4-1 Registers of the clock division control part

Abbreviated Register Name	Register Name	Reference
DIVR0	Divide clock configuration register 0	5.4.1
DIVR1	Divide clock configuration register 1	5.4.2
DIVR2	Divide clock configuration register 2	5.4.3

5.4.1 Divide Clock Configuration Register 0 (DIVR0)

This register sets the source clock (SRCCLK) division rate for generating the base clock (BCLK).

Figure 5.4-1 shows the bit configuration of divide clock configuration register 0 (DIVR0).

Figure 5.4-1 Bit configuration of divide clock configuration register 0 (DIVR0)

bit	7	6	5	4	3	2	1	0
	DIVB2	DIVB1	DIVB0	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	1

R/W: Read/Write

[bit7 to bit5]: DIVB2 to DIVB0 (base clock division configuration bits)

These bits set the division rate for generating the base clock (BCLK) from the source clock (SRCCLK).

Since the CPU clock (CCLK) and the on-chip bus clock (HCLK) are generated without dividing the base clock (BCLK), the frequency is the same as that for the base clock (BCLK).

DIVB2	DIVB1	DIVB0	Explanation
0	0	0	Divided by 1 (undivided)
0	0	1	Divided by 2
0	1	0	Divided by 3
0	1	1	Divided by 4
1	0	0	Divided by 5
1	0	1	Divided by 6
1	1	0	Divided by 7
1	1	1	Divided by 8

[bit4 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: Reserved bits

In case of writing	Always write "1" to this (these) bit (bits)
In case of reading	"1" is read.

5.4.2 Divide Clock Configuration Register 1 (DIVR1)

This register sets the base clock (BCLK) division rate for generating the external bus clock (TCLK). It also controls the stopping of the external bus clock (TCLK).

Figure 5.4-2 shows the bit configuration of divide clock configuration register 1 (DIVR1).

Figure 5.4-2 Bit configuration of divide clock configuration register 1 (DIVR1)

bit	7	6	5	4	3	2	1	0
	TSTP	DIVT2	DIVT1	DIVT0	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	0	0	0	0

R/W: Read/Write

[bit7]: TSTP (External bus clock stop enable bit)

This bit specifies whether to stop the external bus clock (TCLK) when the on-chip bus is stopped in sleep mode.

If such stopping is enabled, the external bus clock (TCLK) is not supplied except at the bus access time.

Written Value	Explanation
0	Do not stop the external bus clock (TCLK).
1	Stop the external bus clock (TCLK).

[bit6 to bit4]: DIVT2 to DIVT0 (External bus clock division configuration bits)

These bits set the division rate for generating the external bus clock (TCLK) from the base clock (BCLK).

DIVT2	DIVT1	DIVT0	Explanation
0	0	0	Divided by 1 (undivided)
0	0	1	Divided by 2
0	1	0	Divided by 3
0	1	1	Divided by 4
1	0	0	Divided by 5
1	0	1	Divided by 6
1	1	0	Divided by 7
1	1	1	Divided by 8

[bit3 to bit0]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

5.4.3 Divide Clock Configuration Register 2 (DIVR2)

This register sets the base clock (BCLK) division rate for generating the peripheral clock (PCLK).

Figure 5.4-3 shows the bit configuration of divide clock configuration register 2 (DIVR2).

Figure 5.4-3 Bit configuration of divide clock configuration register 2 (DIVR2)

bit	7	6	5	4	3	2	1	0
	DIVP3	DIVP2	DIVP1	DIVP0	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	1	0	0	0	0

R/W: Read/Write

[bit7 to bit4]: DIVP3 to DIVP0 (Peripheral clock division configuration bits)

These bits set the division rate for generating the peripheral clock (PCLK) from the base clock (BCLK).

DIVP3	DIVP2	DIVP1	DIVP0	Explanation
0	0	0	0	Divided by 1 (undivided)
0	0	0	1	Divided by 2
0	0	1	0	Divided by 3
0	0	1	1	Divided by 4
0	1	0	0	Divided by 5
0	1	0	1	Divided by 6
0	1	1	0	Divided by 7
0	1	1	1	Divided by 8
1	0	0	0	Divided by 9
1	0	0	1	Divided by 10
1	0	1	0	Divided by 11
1	0	1	1	Divided by 12
1	1	0	0	Divided by 13
1	1	0	1	Divided by 14
1	1	1	0	Divided by 15
1	1	1	1	Divided by 16

[bit3 to bit0]: Reserved bits

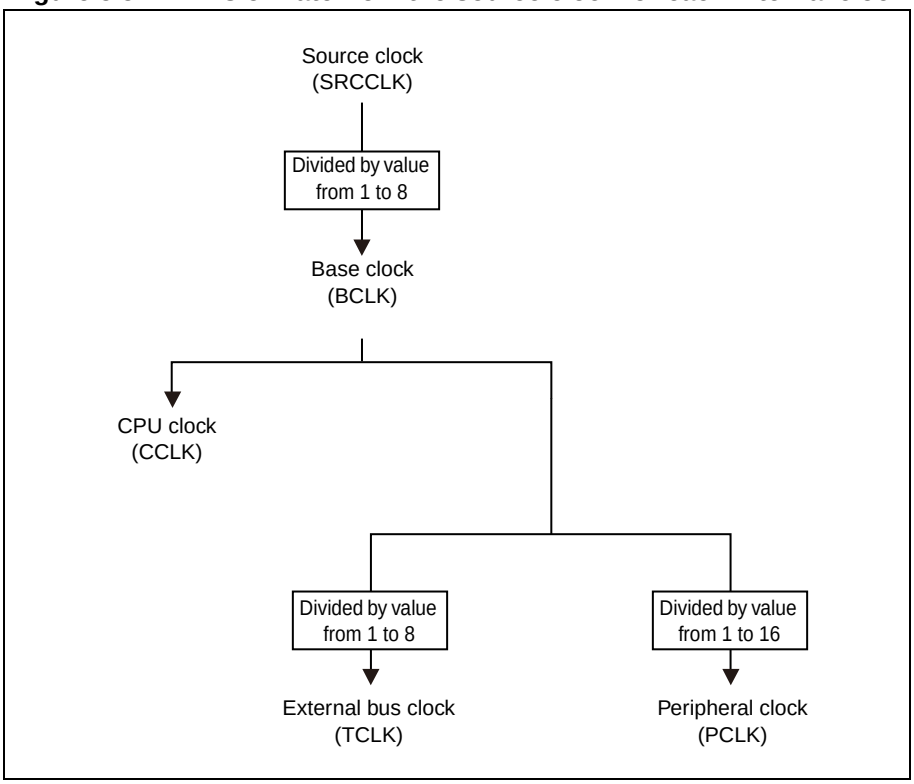
In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is red.

5.5 Division Rate

The clock division control part can set the division rate for each internal clock.

Figure 5.5-1 shows the division rate from the source clock for each internal clock.

Figure 5.5-1 Division rate from the source clock for each internal clock



■ Division rates after initialization

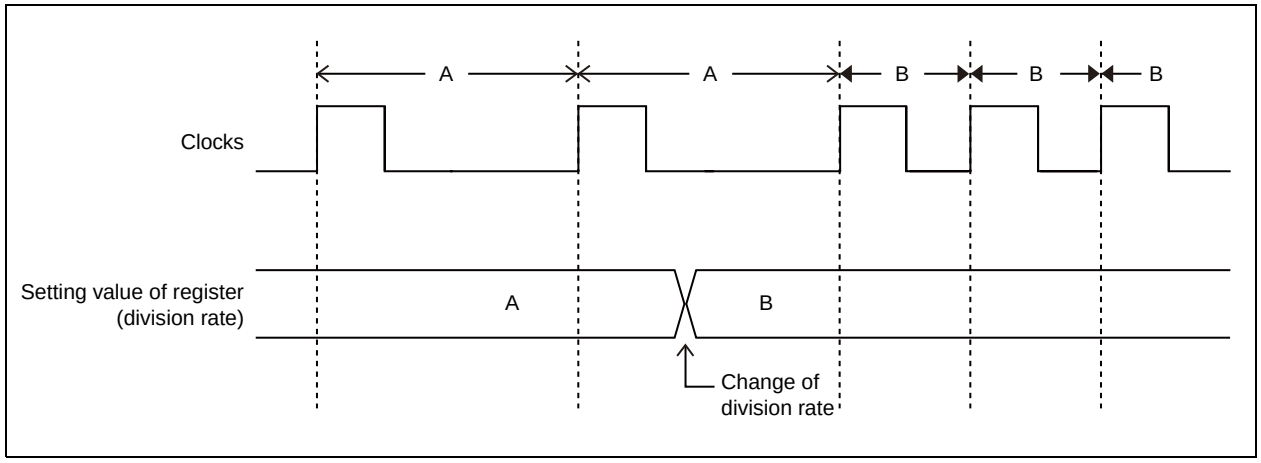
Table 5.5-1 shows the division of internal clocks after a reset.

Table 5.5-1 Division rates after a reset

Clock Name	Division Rate after Initialization
Base clock (BCLK)	Source clock (SRCCLK) divided by 1
CPU clock (CCLK)	Base clock (BCLK) divided by 1
On-chip bus clock (HCLK)	Base clock (BCLK) divided by 1
External bus clock (TCLK)	Base clock (BCLK) divided by 2
Peripheral clock (PCLK)	Base clock (BCLK) divided by 4

■ Changing the division rate

After the division rate setting is changed, the changed division rate is enabled at the next rising edge of the clock.



CHAPTER 6 Main Timer

This chapter explains the functions and operations of the main timer function.

6.1 Overview

6.2 Configuration

6.3 Registers

6.4 Interrupts

6.5 An Explanation of Operations and Setting Procedure Examples

6.1 Overview

The main timer operates with the main clock (MCLK).
The main timer is used to generate the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

The main timer counts the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

When main clock (MCLK) oscillation is stable, the main timer can also be used as an interval timer for generating an interrupt request at regular intervals.

The main timer is cleared when:

- "1" is written to the MTC bit of the main timer control register (MTMCR).

  "1" is read from the MTC bit of the main timer control register (MTMCR) until the main timer is cleared after "1" is written to the MTC bit.
- Main clock (MCLK) oscillation is stopped.

  (The MCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The main timer is stopped with the MTE bit (MTE = 0) of the main timer control register (MTMCR).

If main timer operation is disabled, the timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

6.2 Configuration

This section explains the main timer configuration.

■ Main timer block diagram

For the main timer block diagram, see "■ Main clock (MCLK) generating part" in "CHAPTER 4 Clock Generating Parts".

■ Clocks

Table 6.2-1 shows the clocks used by the main timer.

Table 6.2-1 Clocks used by the main timer

Clock Name	Description
Operation clock	Main clock (MCLK)

6.3 Registers

This section explains the configuration and functions of registers used by the main timer.

■ Registers of main timer

Table 6.3-1 shows the registers used by the main timer.

Table 6.3-1 Main timer registers

Abbreviated Register Name	Register Name	Reference
MTMCR	Main timer control register	6.3.1

6.3.1 Main Timer Control Register (MTMCR)

This register controls the main timer.

Figure 6.3-1 shows the bit configuration of the main timer control register (MTMCR).

Figure 6.3-1 Bit configuration of main timer control register (MTMCR)

bit	7	6	5	4	3	2	1	0
	MTIF	MTIE	MTC	MTE	MTS3	MTS2	MTS1	MTS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	1	1	1	1

R/W: Read/Write

<Notes>

- This register can be rewritten only when the main clock (MCLK) is oscillating stably (The MCRDY bit of the clock source monitor register (CMONR) is 1).

  Note that the MTIE bit can be rewritten even when the MCRDY bit is "0".
- Software reset must be executed when both the MTE and MTC bits are "0". For details of the software reset, see "CHAPTER 9 Reset".

[bit7]: MTIF (main timer interrupt flag bit)

This flag indicates that the main timer overflows.

The main timer overflows when:

- The counter has finished counting the period that is set with the MTS3 to MTS0 bits.
- The oscillation stabilization wait time of the main clock (MCLK) has elapsed after the MCEN bit of the clock source select register (CSELR) was rewritten from "0" to "1".
- The oscillation stabilization wait time of the main clock (MCLK) has elapsed after the system returns from stop mode.

A main timer interrupt request occurs when this bit is set to "1" while the MTIE bit is "1".

MTIF	In case of reading	In case of writing
0	No overflow occurred.	This bit is cleared to "0".
1	An overflow occurred.	Ignored

This bit is also cleared to "0" when a DMA transfer is caused by a main timer interrupt request.

<Notes>

- Disabling main timer operation with the MTE bit (MTE = 0) clears the main timer.
- When the MTIE bit is set to "0", this bit is not cleared even when a DMA transfer is caused by a main timer interrupt request.
- After this device is reset by input of an "L" level signal from the $\overline{\text{INIT}}$ pin, an "H" level signal may be input again from the $\overline{\text{INIT}}$ pin. In this case, this bit is not changed to "1" even after the oscillation stabilization wait time of the main clock (MCLK) elapses.
- If clearing the bit to "0" coincides with the occurrence of an overflow, the overflow occurrence is given priority and this bit remains "1".
- When a read-modify-write instruction is used, "1" is read.

[bit6]: MTIE (main timer interrupt enable bit)

The MTIE bit is used to specify whether to cause a main timer interrupt request when the main timer overflows (MTIF=1).

A main timer interrupt request occurs when the MTIF bit is set to "1" while this bit is "1".

Written Value	Explanation
0	Disables generation of main timer interrupt requests.
1	Enables generation of main timer interrupt requests.

[bit5]: MTC (main timer clear bit)

Clear the main timer.

The operating state of the main timer can be verified by reading this bit.

MTC	In case of writing	In case of reading
0	Ignored	In normal operation
1	Clear the main timer.	The main timer is being cleared.

<Notes>

- When a read-modify-write instruction is used, "0" is read.
- Do not clear the main timer during oscillation stabilization wait time of the PLL clock (PLLCLK).
- This register can be rewritten only while main clock (MCLK) oscillation is stable. Therefore, if the following conditions are satisfied, the main timer cannot be cleared even when the bit is set to "1":
  - Main clock (MCLK) is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
  - The main clock (MCLK) is in oscillation stopped/oscillation stabilization wait state (The MCRDY bit of the clock source monitor register (CMONR) is 0).
- Writing "1" to this bit at the same time that the MTE bit is changed from "0" to "1" clears the main timer and then starts main timer operation.
- Do not write "1" to this bit when it is "1".
- As long as the MTC bit is "0", the MTIF bit may become "1".

[bit4]: MTE (main timer operation enable bit)

This bit enables/disables (stops) the operation of the main timer.

Written Value	Explanation
0	Disables (stops) the operation of the main timer.
1	Enables the operation of the main timer.

<Notes>

- If the operation of the main timer is disabled (stopped), the main timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).
- Disabling (stopping) the operation of the main timer clears the main timer. While the main timer is cleared, "1" is read from the MTC bit. As long as the MTC bit is "0", the MTIF bit may become "1".
- Do not change this bit from "1" to "0" during oscillation stabilization wait time of the PLL clock (PLLCLK).
- Do not write "1" to this bit when the MTC bit is "1".

[bit3 to bit0]: MTS3 to MTS0 (main timer period select bits)

These bits are used to select an overflow period of the main timer.

The main timer overflows when it finishes counting the period specified with these bits.

MTS3	MTS2	MTS1	MTS0	Overflow Period	4 MHz	8 MHz	48 MHz
1	0	0	0	$2^9 \times$ Main clock cycle	128.0 µs	64.0 µs	About 10.7 µs
1	0	0	1	$2^{10} \times$ Main clock cycle	256.0 µs	128.0 µs	About 21.3 µs
1	0	1	0	$2^{11} \times$ Main clock cycle	512.0 µs	256.0 µs	About 42.7 µs
1	0	1	1	$2^{12} \times$ Main clock cycle	About 1 ms	512.0 µs	About 85.3 µs
1	1	0	0	$2^{13} \times$ Main clock cycle	About 2 ms	About 1 ms	About 170.7 µs
1	1	0	1	$2^{14} \times$ Main clock cycle	About 4 ms	About 2 ms	About 341.3 µs
1	1	1	0	$2^{15} \times$ Main clock cycle	About 8 ms	About 4 ms	About 682.7 µs
1	1	1	1	$2^{16} \times$ Main clock cycle	About 16.4 ms	About 8 ms	About 1.4 ms

Always write "1" to the MTS3 bit.

<Notes>

- Change the values of these bits after stopping the main timer using the MTE bit (MTE = 0).
- While the MTIE bit is set to "1", a main timer interrupt request is generated when the main timer overflows.

  Set these bits so that the main timer overflow period exceeds 5T (T: peripheral clock (PCLK) period).

6.4 Interrupts

A main timer interrupt request is generated when the main timer overflows.

Table 6.4-1 outlines the interrupts that can be used with the main timer.

Table 6.4-1 Interrupts of the main timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Main timer interrupt request	MTIF=1 for MTMCR	MTIE=1 for MTMCR	Write "0" to the MTIF bit for MTMCR

MTMCR: main timer control register (MTMCR)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For information on the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

6.5 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the main timer. Also, examples of procedures for setting the operating state are shown.

6.5.1 Main Timer Operation

■ Overview

The main timer counts the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

When main clock (MCLK) oscillation is stable, the main timer can also be used as an interval timer for generating an interrupt request at regular intervals.

If main timer operation is disabled with the MTE bit (MTE = 0) of the main timer control register (MTMCR), the timer is stopped during periods other than the oscillation stabilization wait time of the main clock (MCLK) and PLL clock (PLLCLK).

■ Operation

The main timer operates as follows:

1. Enable the main timer operation by the MTE bit of the main timer control register (MTMCR) (MTE = 1).
2. The main timer starts counting in synchronization with the main clock (MCLK).

   The main timer continues counting while the MTE bit of the main timer control register (MTMCR) is "1".
3. The main timer counts up to the value set in the MTS3 to MTS0 bits of the main timer control register (MTMCR).

   The MTIF bit of the main timer control register (MTMCR) changes to "1".

   If the MTIE bit of the main timer control register (MTMCR) is "1" at this time, a main timer interrupt request is generated.

   To clear the main timer interrupt request, write "0" to the MTIF bit. The MTIF bit is cleared to "0".

If main timer operation is disabled with the MTE bit (MTE=0) of the main timer control register (MTMCR) during main timer operation, the main timer stops counting and clears the counter value. For more information, see "■ Clearing the timer".

■ Clearing the timer

The main timer is cleared when:

- "1" is written to the MTC bit of the main timer control register (MTMCR).

  "1" is read from the MTC bit of the main timer control register (MTMCR) until the main timer is cleared after "1" is written to the MTC bit.
- Main clock (MCLK) oscillation is stopped.

  (The MCEN bit of the clock source select register (CSELR) is 0).
- In stop mode
- The main timer is stopped with the MTE bit (MTE = 0) of the main timer control register (MTMCR).

<Note>

The main timer control register (MTMCR) can be rewritten only when the oscillation of the main clock (MCLK) is stable. Therefore, even if "1" is written to the MTC bit of the main timer control register (MTMCR) when the following conditions are satisfied, the main timer cannot be cleared:

- Main clock (MCLK) oscillation is oscillating (the MCEN bit of the clock source select register (CSELR) is 1).
- The main clock (MCLK) is in oscillation stopped/oscillation stabilization wait state (The MCRDY bit of the clock source monitor register (CMONR) is 0).

■ Interrupt setting procedure

An example of the procedure for setting the main timer control register (MTMCR) is shown below.

1. Set the MTIE bit to disable main timer interrupts (MTIE=0).
2. Set the MTIF bit to clear the main timer interrupt flag (MTIF=0).
3. Set the MTE bit to disable main timer operation (MTE=0).
4. Read the MTC bit to verify that the main timer has been cleared (MTC=0).
5. Set the timer period in the MTS3 to MTS0 bits.
6. Set the MTIE bit to enable main timer interrupts (MTIE=1).
7. Set the MTE bit to enable main timer operation (MTE=1).

   When the period that is set in the MTS3 to MTS0 bits elapses, a main timer interrupt request is generated and processing moves to the interrupt processing routine.
8. Set the MTIF bit to clear the main timer interrupt flag (MTIF=0).
9. Read the MTIF bit once to complete clearing the main timer interrupt flag.

   Issue the RETI instruction to return to normal program processing from the interrupt processing routine.

<Note>

When "0" is written to the MTIF bit, the main timer interrupt flag is not cleared soon. After reading the MTIF bit once to complete clearing the flag, it can be returned by the RETI instruction.

6.5.2 Transition to Stop Mode

Before transition to the stop mode, generation of main timer interrupt requests must be disabled.

Follow the procedure below for transition to the stop mode:

1. Set the PCEN bit of the clock source select register (CSELR) to stop PLL clock (PLLCLK) oscillation (PCEN=0).
2. Set the MTIE bit of the main timer control register (MTMCR) to disable generation of main timer interrupt requests (MTIE=0).
3. Set the MTE bit of the main timer control register (MTMCR) to disable main timer operation (MTE = 0).
4. Read the MTC bit of the main timer control register (MTMCR) to verify that the main timer is not being cleared (MTC=0).
5. Set the MTIF bit of the main timer control register (MTMCR) to clear the main timer interrupt flag (MTIF=0).
6. Set the oscillation stabilization wait time of the main clock (MCLK) in the MOSW3 to MOSW0 bits of the clock stabilization time select register (CSTBR).
7. Transition to stop mode

<Note>

Before transition to stop mode, be sure to stop PLL clock (PLLCLK) oscillation.

CHAPTER 7 Sub Timer

This chapter explains the functions and operations of the sub timer.

7.1 Overview

7.2 Configuration

7.3 Registers

7.4 Interrupts

7.5 An Explanation of Operations and Setting Procedure Examples

7.1 Overview

The sub timer operates based on the sub clock (SBCLK).
It is used to generate the sub clock (SBCLK) oscillation stabilization wait time.

The sub timer counts the oscillation stabilization wait time of the sub clock (SBCLK).

When sub clock (SBCLK) oscillation is stable, the sub timer can also be used as an interval timer for generating an interrupt request at regular intervals.

The sub timer is cleared when:

- "1" is written to the STC bit of the sub timer control register (STMCR).

  "1" is read from the STC bit of the sub timer control register (STMCR) until the sub timer is cleared after "1" is written to the STC bit.
- Sub clock (SBCLK) oscillation is stopped.

  (The SCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The sub timer is stopped with the STE bit (STE=0) of the sub timer control register (STMCR).

If sub timer operation is disabled, the timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

7.2 Configuration

This section explains the sub timer configuration.

■ Sub timer block diagram

For details of the sub timer block diagram, see "■ Sub clock (SBCLK) generating part" in "CHAPTER 4 Clock Generating Parts".

■ Clocks

Table 7.2-1 shows the clocks used by the sub timer.

Table 7.2-1 Clocks used by the sub timer

Clock Name	Description
Operation clock	Sub clock (SBCLK)

7.3 Registers

This section explains the configuration and functions of registers used by the sub timer.

■ Registers of sub timer

The registers used by the sub timer are listed in Table 7.3-1.

Table 7.3-1 Sub timer registers

Abbreviated Register Name	Register Name	Reference
STMCR	Sub timer control register	7.3.1

7.3.1 Sub Timer Control Register (STMCR)

This register controls the sub timer.

Figure 7.3-1 shows the bit configuration of the sub timer control register (STMCR).

Figure 7.3-1 Bit configuration of sub timer control register (STMCR)

bit	7	6	5	4	3	2	1	0
	STIF	STIE	STC	STE	Reserved	STS2	STS1	STS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	1	1	1

R/W: Read/Write

<Notes>

- This register can be rewritten only when the sub clock (SBCLK) is oscillating stably. (The SCRDY bit of the clock source monitor register (CMONR) is 1.)

  Note that the STIE bit can be rewritten even when the SCRDY bit is "0".
- Software reset must be executed when both the STE and STC bits are "0". For details of the software reset, see "CHAPTER 9 Reset".

[bit7]: STIF (sub clock timer interrupt flag bit)

This flag indicates that the sub timer caused an overflow.

The sub timer overflows when:

- The counter has finished counting the period that is set with the STS2 to STS0 bits.
- The oscillation stabilization wait time of the sub clock (SBCLK) has elapsed after the SCEN bit of the clock source select register (CSELR) was rewritten from "0" to "1".
- The oscillation stabilization wait time of the sub clock (SBCLK) has elapsed after the system returns from stop mode.

A sub timer interrupt request occurs when this bit is set to "1" while the STIE bit is "1".

STIF	In case of reading	In case of writing
0	No overflow occurred.	This bit is cleared to "0".
1	An overflow occurred.	Ignored

This bit is also cleared to "0" when a DMA transfer is caused by a sub timer interrupt request.

<Notes>

- Disabling sub timer operation with the STE bit (STE = 0) clears the sub timer.
- When the STIE bit is set to "0", this bit is not cleared even when a DMA transfer is caused by a sub timer interrupt request.
- If clearing the bit to "0" coincides with the occurrence of an overflow, the overflow occurrence is given priority and this bit remains "1".
- When a read-modify-write instruction is used, "1" is read

[bit6]: STIE (sub timer interrupt enable bit)

The STIE bit is used to specify whether to cause a sub timer interrupt request when the sub timer overflows (STIF=1).

A sub timer interrupt request occurs when the STIF bit is set to "1" while this bit is "1".

Written Value	Explanation
0	Disables generation of sub timer interrupt requests.
1	Enables generation of sub timer interrupt requests.

[bit5]: STC (sub timer clear bit)

This bit clears the sub timer.

The operating state of the sub timer can be verified by reading this bit.

STC	In case of writing	In case of reading
0	Ignored	In normal operation
1	Clear the sub timer.	The sub timer is being cleared.

<Notes>

- When a read-modify-write instruction is used, "0" is read.
- This register can be rewritten only while sub clock (SBCLK) oscillation is stable. Therefore, if the following conditions are satisfied, the sub timer cannot be cleared even when the bit is set to "1":
  - Sub clock (SBCLK) is oscillating (the SCEN bit of the clock source select register (CSELR) is 1).
  - The sub clock (SBCLK) is in oscillation stopped/oscillation stabilization wait state. (The SCRDY bit of the clock source monitor register (CMONR) is 0.)
- Writing "1" to this bit at the same time that the STE bit is changed from "0" to "1" clears the sub timer and then starts sub timer operation.
- Do not attempt to write "1" to this bit when it is "1".
- As long as the STC bit is "0", the STIF bit may become "1".

[bit4]: STE (sub timer operation enable bit)

This bit controls the sub timer operation.

Written Value	Explanation
0	Disables (stops) the operation of the sub timer.
1	Enables the operation of the sub timer.

<Notes>

- If the operation of the sub timer is disabled (stopped), the sub timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).
- Disabling (stopping) the operation of the sub timer clears the sub timer. While the sub timer is cleared, "1" is read from the STC bit. As long as the STC bit is "0", the STIF bit may become "1".
- Do not write "1" to this bit when the STC bit is "1".

[bit3]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2 to bit0]: STS2 to STS0 (sub timer period select bits)

These bits are used to select an overflow period of the sub timer.

The sub timer overflows when it finishes counting the period specified with these bits.

STS2	STS1	STS0	Overflow Period	At 32768Hz
0	0	0	$2^8 \times$ Sub clock cycle	About 7.8 ms
0	0	1	$2^9 \times$ Sub clock cycle	About 15.6 ms
0	1	0	$2^{10} \times$ Sub clock cycle	About 31.3 ms
0	1	1	$2^{11} \times$ Sub clock cycle	62.5 ms
1	0	0	$2^{12} \times$ Sub clock cycle	125.0 ms
1	0	1	$2^{13} \times$ Sub clock cycle	250.0 ms
1	1	0	$2^{14} \times$ Sub clock cycle	500.0 ms
1	1	1	$2^{15} \times$ Sub clock cycle	1 s

<Notes>

- Change the values of these bits after stopping the sub timer using the STE bit (STE = 0).
- While the STIE bit is set to "1", a sub timer interrupt request is generated when the sub timer overflows.

  Set these bits so that the sub timer overflow period is 5T (T: peripheral clock (PCLK) period) or more than that.

7.4 Interrupts

A sub timer interrupt request is generated when the sub timer overflows.

Table 7.4-1 outlines the interrupts that can be used with the sub timer.

Table 7.4-1 Interrupts of the sub timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Sub timer interrupt request	STIF=1 for STMCR	STIE=1 for STMCR	Write "0" to the STIF bit for STMCR

STMCR: sub timer control register (STMCR)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the interrupt level settings, see "CHAPTER 10 Interrupt Controller".

7.5 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the sub timer. Also, examples of procedures for setting the operating state are shown.

7.5.1 Sub timer operation

■ Overview

The sub timer counts the oscillation stabilization wait time of the sub clock (SBCLK).

When sub clock (SBCLK) oscillation is stable, the sub timer can also be used as an interval timer for generating an interrupt request at regular intervals.

If sub timer operation is disabled with the STE bit (STE = 0) of the sub timer control register (STMCR), the timer is stopped during periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

■ Operation

The sub timer operates as follows:

1. The STE bit of the sub timer control register (STMCR) enables (STE = 1) sub timer operation.
2. The sub timer starts counting in synchronization with the sub clock (SBCLK).

   The sub timer continues counting while the STE bit of the sub timer control register (STMCR) is "1".
3. The sub timer counts up to the value specified in the STS2 to STS0 bits of the sub timer control register (STMCR).

   The STIF bit of the sub timer control register (STMCR) changes to "1".

   If the STIE bit of the sub timer control register (STMCR) is "1" at this time, a sub timer interrupt request is generated.

   To clear the sub timer interrupt request, write "0" to the STIF bit. The STIF bit is cleared to "0".

If sub timer operation is disabled with the STE bit (STE = 0) of the sub timer control register (STMCR) during sub timer operation, the sub timer stops counting and clears the counter value. For more information, see "■ Clearing the timer".

■ Clearing the timer

The sub timer is cleared when:

- "1" is written to the STC bit of the sub timer control register (STMCR).

  "1" is read from the STC bit of the sub timer control register (STMCR) until the sub timer is cleared after "1" is written to the STC bit.
- Sub clock (SBCLK) oscillation is stopped.

  (The SCEN bit of the clock source select register (CSELR) is 0.)
- In stop mode
- The sub timer is stopped with the STE bit (STE = 0) of the sub timer control register (STMCR).

  The sub timer is stopped for periods other than the oscillation stabilization wait time of the sub clock (SBCLK).

<Note>

The sub timer control register (STMCR) can be rewritten only while the oscillation of the sub clock (SBCLK) is stable. Therefore, even if "1" is written to the STC bit of the sub timer control register (STMCR) when the following conditions are satisfied, the sub timer cannot be cleared:

- Sub clock (SBCLK) is oscillating. (The SCEN bit of the clock source select register (CSELR) is 1.)
- The sub clock (SBCLK) is in oscillation stopped/oscillation stabilization wait state. (The SCRDY bit of the clock source monitor register (CMONR) is 0.)

■ Interrupt setting procedure

An example of the procedure for setting the sub timer control register (STMCR) is shown below.

1. Set the STIE bit to disable sub timer interrupts (STIE = 0).
2. Set the STIF bit to clear the sub timer interrupt flag (STIF = 0).
3. Set the STE bit to disable sub timer operation (STE = 0).
4. Read the STC bit to verify that the sub timer is operating normally (STC=0).
5. Set the timer period in the STS2 to STS0 bits.
6. Set the STIE bit to enable sub timer interrupts (STIE = 1).
7. Set the STE bit to enable sub timer operation (STE = 1).

   When the period that is set in the STS2 to STS0 bits elapses, a sub timer interrupt request is generated and processing moves to the interrupt processing routine.
8. Set the STIF bit to clear the sub timer interrupt flag (STIF = 0).
9. Read the STIF bit once to complete clearing the sub timer interrupt flag.

   Issue the RETI instruction to return to normal program processing from the interrupt processing routine.

<Note>

When "0" is written to the STIF bit, the sub timer interrupt flag is not cleared soon. After reading the STIF bit once to complete clearing the flag, it can be returned by the RETI instruction.

7.5.2 Transition to Stop Mode, and Watch Mode

Before transition to stop mode, interrupt operation by the sub timer must be disabled.

Follow the procedure below for transition to the stop mode:

1. Set the PCEN bit of the clock source select register (CSELR) to stop PLL clock (PLLCLK) oscillation (PCEN=0).
2. Set the STIE bit of the sub timer control register (STMCR) to disable sub timer interrupts (STIE = 0).
3. Set the STE bit of the sub timer control register (STMCR) to disable sub timer operation (STE = 0).
4. Read the STC bit of the sub timer control register (STMCR) to confirm that the sub timer is not being cleared (STC=0).
5. Set the STIF bit of the sub timer control register (STMCR) to clear the sub timer interrupt flag (STIF = 0).
6. Set the oscillation stabilization wait time of the sub clock (SBCLK) in the SOSW2 to SOSW0 bits of the clock stabilization time select register (CSTBR).
7. Transition to stop mode

<Note>

Before transition to the stop mode, be sure to stop PLL clock oscillation.

CHAPTER 8 Low-power Dissipation Mode

This chapter explains the functions and operations of low-power dissipation mode.

8.1 Overview

8.2 Configuration

8.3 Registers

8.4 An Explanation of Operations and Setting Procedure Examples

8.5 Notes on Use

8.1 Overview

This series can use low-power dissipation mode to reduce power dissipation.

■ Overview

This series can control power dissipation in the following way.

- Clock control
  - Clock division
    By changing the division ratio of each operation clock, operation frequency can be reduced.
  - Stop clock
    This allows the user to specify a specific clock to stop the clock.
- Doze mode
  This mode intermittently operates the CPU repeatedly at a set operation rate.
- Sleep mode
  This mode operates only peripheral functions. One of the following two modes can be selected.
  - CPU sleep mode
    This mode stops the operation of the CPU.
  - Bus sleep mode
    This mode stops the CPU and on-chip bus.
- Standby mode
  One of the following three modes can be selected.
  - Main timer mode
    This mode stops all the operations other than the main clock oscillation.
    The sub clock oscillation can be specified arbitrarily.
  - Watch mode
    This mode stops all the operations other than the sub clock oscillation.
  - Stop mode
    This mode stops all operations including the oscillation of all clocks.

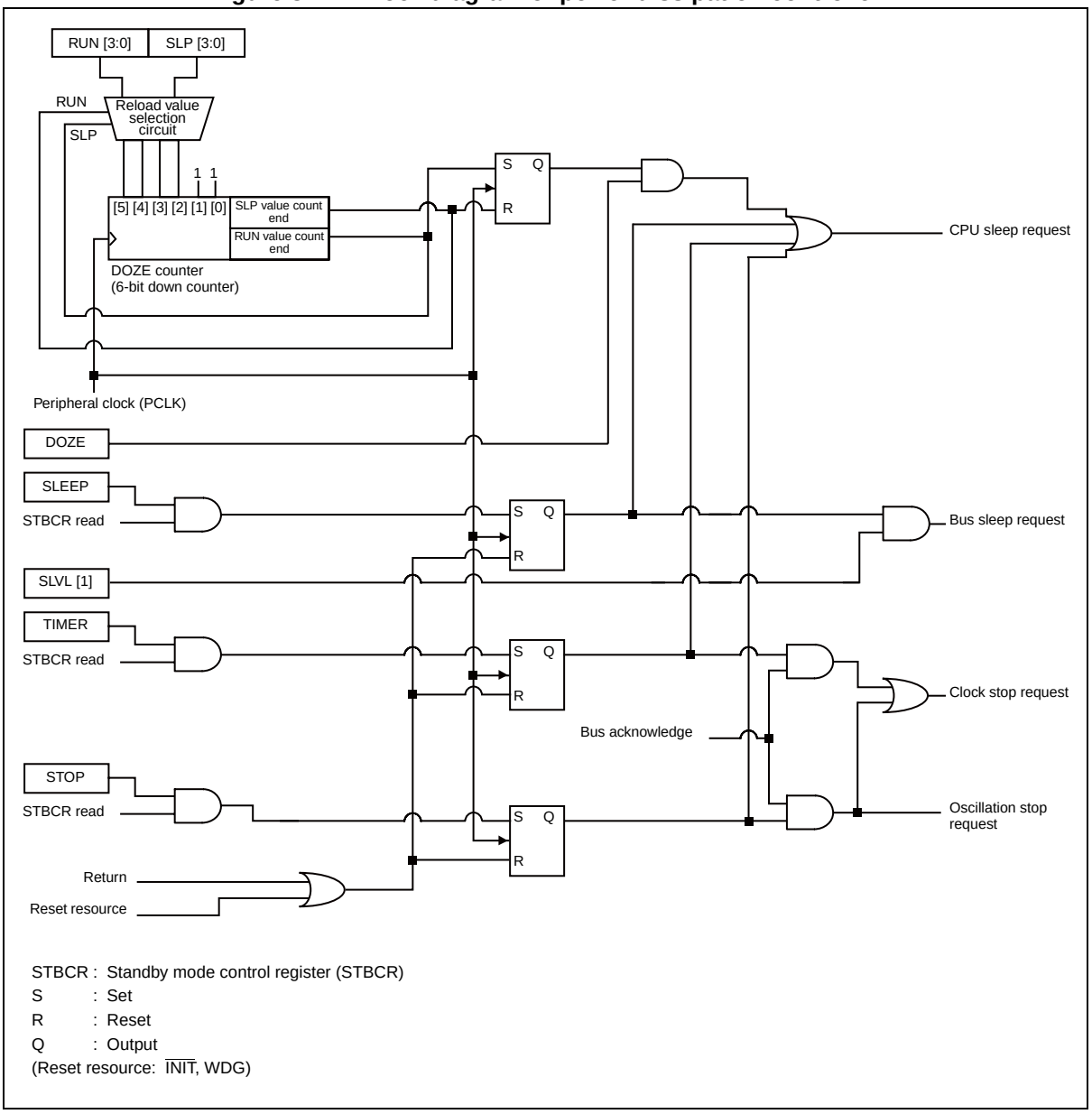
8.2 Configuration

The configuration of the power dissipation controller is shown below.

■ Block diagram of power dissipation controller

Figure 8.2-1 is a block diagram of the power dissipation controller.

Figure 8.2-1 Block diagram of power dissipation controller



- Standby mode control register (STBCR)

  This register controls low-power dissipation mode.
- Sleep rate configuration register (SLPRR)

  This register configures the operation state (RUN state) rate and sleep state rate (sleep rate) in doze mode.

- Reload value selection circuit

  A circuit for selecting to reload either the operation state (RUN state) rate or sleep state rate (Sleep rate) which has been set in the sleep rate configuration register (SLPRR).

■ Clocks

Table 8.2-1 shows the clock used in the power dissipation controller.

Table 8.2-1 Clock used in power dissipation controller

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-

8.3 Registers

This section explains the configurations and functions of the registers that are required for controlling power dissipation.

■ List of registers that control power dissipation

Table 8.3-1 is a list of registers that control power dissipation.

Table 8.3-1 List of registers that control power dissipation

Abbreviated Register Name	Register Name	Reference
STBCR	Standby mode control register	8.3.1
SLPRR	Sleep rate configuration register	8.3.2

8.3.1 Standby Mode Control Register (STBCR)

This register controls low-power dissipation mode.

Figure 8.3-1 shows the bit configuration of the standby mode control register (STBCR).

Figure 8.3-1 Bit configuration of the standby mode control register (STBCR)

bit	7	6	5	4	3	2	1	0
	STOP	TIMER	SLEEP	DOZE	Reserved	Reserved	SLVL1	SLVL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	1	1

R/W: Read/Write

[bit7]: STOP (Stop mode enable bit)

This bit enables transition to stop mode.

Written Value	Explanation
0	Does not transit to stop mode.
1	Transits to stop mode.

If this register is read after this bit enables transition to stop mode, power dissipation mode moves to stop mode.

If the return resource from stop mode occurs, this bit is cleared to "0". For information on return resource from stop mode, see "■ Return from stop mode" in "8.4.6 Operation in Stop Mode".

[bit6]: TIMER (Main timer mode/watch mode enable bit)

This bit enables transition to main timer mode/watch mode.

Written Value	Explanation
0	Does not transit to main timer mode/watch mode.
1	Transits to main timer mode/watch mode.

If this register is read after this bit enables transition to main timer mode/watch mode, power dissipation mode moves to main timer mode/watch mode.

If, however, transition to stop mode is enabled with the STOP bit (STOP = 1), the setting of this bit is ignored even when transition to main timer mode/watch mode is enabled by writing "1" to this bit.

If the return resource from main timer mode/watch mode occurs, this bit is cleared to "0". For information on return resource from main timer mode, see "■ Return from the main timer mode" in "8.4.4 Operation in Main Timer Mode". For information on return resource from watch mode, see "■ Return from the watch mode" in "8.4.5 Operation in Watch Mode".

[bit5]: SLEEP (Sleep mode enable bit)

This bit enables transition to sleep mode.

Written Value	Explanation
0	Does not transit to sleep mode.
1	Transits to sleep mode.

If this register is read after this bit enables transition to sleep mode, power dissipation mode moves to sleep mode.

If, however, transition to stop mode/main timer mode/watch mode is enabled with the STOP bit/TIMER bit (STOP/TIMER = 1), the setting of this bit is ignored even when transition to sleep mode is enabled by writing "1" to this bit.

If the return resource from sleep mode occurs, this bit is cleared to "0". For information on return resource from sleep mode, see "■ Return from sleep mode" in "8.4.3 Operation in Sleep Mode".

[bit4]: DOZE (Doze mode enable bit)

This bit enables transition to doze mode.

Written Value	Explanation
0	Does not transit to doze mode (CPU intermittent sleep).
1	The CPU transits to doze mode (CPU intermittent sleep).

While the SLVL1 bit is set to "0", if the return resource from doze mode occurs, this bit is cleared to "0". For information on return resource from doze mode, see "■ Return from doze mode" in "8.4.2 Operation in Doze Mode".

[bit3, bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1, bit0]: SLVL1, SLVL0 (Standby level setting bits)

The meaning of the value to be written to this bit varies depending on the low-power dissipation mode to move to.

Low-power Dissipation Mode	SLVL1	SLVL0	Explanation
Stop mode/ Main timer mode/ watch mode	0	0	Does not place the output from each pin in Hi-Z in stop mode/main timer mode/watch mode.
	0	1	
	1	0	Places the output from each pin in Hi-Z in stop mode/ main timer mode/watch mode.
	1	1	
Sleep mode	0	0	When moving to sleep mode, power dissipation mode moves to CPU sleep mode (stops only the operation of the CPU).
	0	1	
	1	0	When moving to sleep mode, power dissipation mode moves to bus sleep mode (stops operations of the CPU and on-chip bus). *
	1	1	
Doze mode	0	0	When interrupt request occur, the DOZE bit is cleared to "0".
	0	1	
	1	0	When interrupt request occur, the DOZE bit is not cleared to "0".
	1	1	

* During DMA transfer, the on-chip bus operates.

<Note>

For information on pins of which the output can be placed in Hi-Z in stop mode/main timer mode/ watch mode, see "APPENDIX D Pin State in Each CPU State".

8.3.2 Sleep Rate Configuration Register (SLPRR)

This register configures the operation state (RUN state) rate and sleep state rate (sleep rate) in doze mode.

Figure 8.3-2 shows the bit configuration of the sleep rate configuration register (SLPRR).

Figure 8.3-2 Bit configuration of the sleep rate configuration register (SLPRR)

bit	7	6	5	4	3	2	1	0
	RUN3	RUN2	RUN1	RUN0	SLP3	SLP2	SLP1	SLP0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

If this register is rewritten in doze mode, the rewritten setting is reflected at the next stop/activation timing.

[bit7 to bit4]: RUN3 to RUN0 (Operation period bits)

These bits set the period during which the CPU operates in doze mode.

The CPU operation period is calculated from the value that is set to these bits as follows.

$$\textbf{(Value of this bit + 1)} \times \mathbf{4} \times \mathbf{t_{CYCP}}$$

$\mathbf{t_{CYCP}}$: Period of the peripheral clock (PCLK)

For details of operation period, see "8.4.2 Operation in Doze Mode".

[bit3 to bit0]: SLP3 to SLP0 (Sleep state period bits)

These bits set the period of sleep state in doze mode.

The sleep state period is calculated from the value that is set to these bits as follows.

$$\textbf{(Value of this bit + 1)} \times 4 \times t_{CYCP}$$

t_{CYCP} : Period of the peripheral clock (PCLK)

For details of the sleep state period, see "8.4.2 Operation in Doze Mode".

<Notes>

- A delay may occur when the CPU accepts the sleep request. In this case, the sleep period will be shorter than that obtained from the above calculation formula.
- If the sleep state period is short, the CPU may not enter the sleep state depending on the operating status of the CPU.

8.4 An Explanation of Operations and Setting Procedure Examples

This section explains the operation and use of low-power dissipation mode and includes examples of the procedure for setting this mode.

■ Overview

You can reduce power dissipation by changing the division ratio of the operation clock or stopping the operation clock.

You can also use the following low-power dissipation modes:

- Doze mode

  This mode intermittently operates the CPU repeatedly at a set operation rate.

  By repeating operation and stop of the CPU alternately in the set period, the average power dissipation of the CPU can be reduced.

- Sleep mode

  In this mode, only the peripheral functions operate while the CPU and on-chip bus are stopped.

  One of the following two modes can be selected.

  - CPU sleep mode

    This mode stops the operation of the CPU.

  - Bus sleep mode

    This mode stops the CPU and on-chip bus.

- Standby mode

  This mode stops the entire device to put it in a standby state.

  One of the following three modes can be selected.

  - Main timer mode
  - Watch mode
  - Stop mode

8.4.1 Operation When Clock Control Is Set

Power dissipation and CPU performance can be optimized by adjusting the operation clocks that are built in this series.

■ Overview

To reduce power dissipation by controlling the clock, the following two methods are available.

- Clock division

  By changing the division ratio of each operation clock, the operation frequency can be reduced.
- Stop clock

  This allows the user to specify a specific clock to stop.

■ Clock division

By changing the division ratio of each operation clock, power dissipation can be reduced. The division ratio of the operation clock can be individually set.

Table 8.4-1 shows each operation clock and settable division ratio.

Table 8.4-1 Operation clock and settable division ratio

Operation Clock	Division Ratio
Base clock (BCLK)	Source clock (SRCCLK) divided by 1 to 8.
External bus clock (TCLK)	Base clock (BCLK) divided by 1 to 8.
Peripheral clock (PCLK)	Base clock (BCLK) divided by 1 to 16.

<Note>

The division method or condition differs depending on the operation clock. For information on the division of the operation clock, see "CHAPTER 5 Clock Division Control Part".

■ Stopping the clock

You can reduce power dissipation by stopping the unused operation clock.

Table 8.4-2 shows the relationship between the operation clock that can be stopped and the deliver/stop timing.

Table 8.4-2 Relationship between the operation clock that can be stopped and the deliver/stop timing

Operation Clock	Deliver/Stop Timing
External bus clock (TCLK)	Bus in sleep mode

Enabling the stop of the external bus clock (TCLK) automatically disables the external bus clock (TCLK) delivery during the period in which there is no access by using the external bus.

If an access is attempted, clock delivery is resumed automatically and delivery is disabled again after access is completed. For information on conditions for disabling external bus clock (TCLK), see "CHAPTER 5 Clock Division Control Part".

8.4.2 Operation in Doze Mode

This mode intermittently operates the CPU in order to reduce the average power dissipation by the CPU.

■ Overview

Using doze mode enables reducing the average power dissipation by the CPU by operating and stopping the CPU alternately at a set interval. Maintain performance while reducing power dissipation by changing the sleep rate according to the processing load.

■ Setting the period

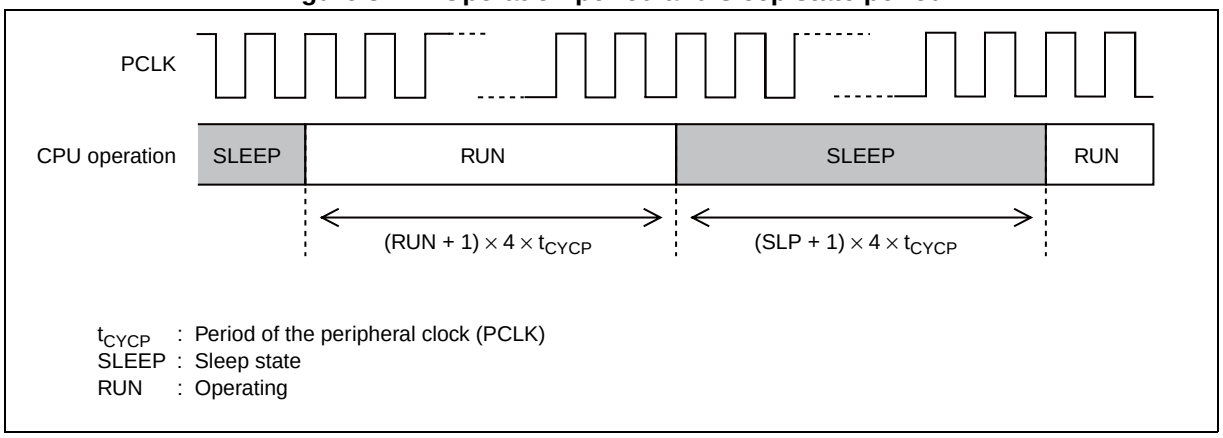
If you set the CPU operation period in the RUN3 to RUN0 bits and sleep state period in the SLP3 to SLP0 bits of the sleep rate configuration register (SLPRR), the period will be calculated from the set value using the following calculation formula.

$$\mathbf{(RUN + 1) \times 4 \times t_{CYCP} + (SLP + 1) \times 4 \times t_{CYCP}}$$

RUN: Value for the RUN3 to RUN0 bits
SLP: Value for the SLP3 to SLP0 bits
$\mathbf{t_{CYCP}}$: Period of the peripheral clock (PCLK)

Figure 8.4-1 shows each cycle.

Figure 8.4-1 Operation period and sleep state period



<Notes>

- The above calculation formula does not contain delay time for the CPU to accept the sleep request. Therefore an error may occur.
- If the setting of the sleep state period is short, the CPU may not enter the sleep state depending on the operating status of the CPU.

■ Transition

If "1" is written to the DOZE bit in the standby mode control register (STBCR) after the cycle is set, doze mode is entered and the CPU starts intermittent operation by alternately running and stopping according to the setting configured in the sleep rate configuration register (SLPRR).

To return from doze mode, write "0" in the DOZE bit of standby mode control register (STBCR).

<Note>

If the sleep rate configuration register (SLPRR) is rewritten in doze mode, the rewritten setting is reflected at the next stop/operation transition timing.

■ Return from doze mode

The CPU returns from doze mode in either of the following cases.

- This device is reset.
- "0" is written to the DOZE bit of standby mode control register (STBCR).
- An interrupt request is generated when the SLVL1 bit of standby mode control register (STBCR) is "0".

Except the above cases, the configuration is retained so that you can use doze mode even after returning from sleep mode, main timer mode, watch mode, or stop mode.

8.4.3 Operation in Sleep Mode

This mode is used to reduce power dissipation in the event wait state.
If sleep mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

■ Overview

Using sleep mode can significantly reduce power dissipation in the event wait state by stopping the CPU and on-chip bus while allowing only the peripheral functions to operate.

The following two modes are available for sleep mode.

- CPU sleep mode

  This mode stops only the operation of the CPU.

  Because the clock continues to be delivered to the DMA controller (DMAC) or to the on-chip bus, operations of these devices continue.

  Though the power dissipation is larger than that in bus sleep mode, quick response can be given to the DMA transfer request.

- Bus sleep mode

  This mode stops the operation of the CPU and on-chip bus.

  It also disables the clock delivery to the DMAC controller (DMAC) or on-chip bus. For information on disabling clock, see "CHAPTER 5 Clock Division Control Part".

  However, if the DMA transfer request is accepted, the clock delivery to the DMA controller (DMAC) or on-chip bus will be tentatively resumed to allow DMA transfer.

  After the DMA transfer is completed, the clock delivery will be disabled again.

You can set whether to disable external bus clock (TCLK) delivery in bus sleep mode, using the TSTP bit in the divide clock configuration register 1 (DIVR1).

For information on the divide clock configuration register 1 (DIVR1), see "5.4.2 Divide Clock Configuration Register 1 (DIVR1)".

While this mode is slower in responding to the DMA transfer request than in CPU sleep mode, it can reduce power dissipation.

■ Setting

Table 8.4-3 shows the settings required before changing to sleep mode.

Table 8.4-3 Setting register

Registers	Bit	Explanation
Divide clock configuration register 1 (DIVR1)	TSTP	Sets whether to enable the external bus clock (TCLK) delivery 0 = Enabling 1 = Disabling
Standby mode control register (STBCR)	SLVL1	Sets whether to change to CPU sleep mode or to bus sleep mode 0 = Change to CPU sleep mode 1 = Change to bus sleep mode

<Note>

If the external bus clock (TCLK) delivery is disabled by setting the TSTP bit (TSTP =1) in divide clock configuration register 1 (DIVR1), DMA transfer cannot be activated by the external DMA transfer request.

■ Transition

By following the steps below, power dissipation mode moves to sleep mode.

1. Write "0" to the STOP bit, write "0" to the TIMER bit, and write "1" to the SLEEP bit of standby mode control register (STBCR).
2. Read standby mode control register (STBCR).

<Note>

To prevent the CPU from executing the next instruction before moving to sleep mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_sleep, R0     ; SLEEP bit=1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to sleep mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

■ Return from sleep mode

The CPU returns from sleep mode in either of the following cases.

- This device is reset.
- An interrupt request is generated (whose interrupt level is other than "31").

  For information on the interrupt level, see "CHAPTER 10 Interrupt Controller".

<Notes>

- If the interrupt request is not accepted by the CPU when returning from sleep mode due to the interrupt request, the program is executed starting from the next instruction after entering sleep mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.
- In bus sleep mode, if a DMA transfer request is generated, the on-chip bus clock (HCLK) delivery is tentatively resumed to perform DMA transfer. The on-chip bus clock (HCLK) delivery is again disabled after DMA transfer is completed.

8.4.4 Operation in Main Timer Mode

Main timer mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state. The permitted clock oscillation, however, operates, allowing less reduction in power dissipation than in stop mode.
In main timer mode, select the main clock (MCLK) oscillation as a source clock (SRCCLK) for the CPU.
If main timer mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

■ Overview

In main timer mode, because main clock (MCLK) oscillation is permitted as a source clock (SRCCLK) for the CPU, the count operation of the main timer is executed.

The sub clock (SBCLK) oscillation can be specified arbitrarily.

■ Setting

Table 8.4-4 shows the settings required before changing to main timer mode.

Table 8.4-4 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects main clock (MCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=00 or 01)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
	SCEN	Specify sub clock (SBCLK) oscillation. 0=Stop oscillation 1=Start oscillation
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in main timer mode 0 = Retain the state in effect before main timer mode is entered 1 = Hi-Z

<Note>

When moving to main timer mode, if the SLVL1 bit of the standby mode control register (STBCR) is set to "0" while setting doze mode, the DOZE bit is cleared to "0" on returning from main timer mode to end doze mode.

■ Transition

By following the steps below, power dissipation mode moves to main timer mode.

1. Write "0" to the STOP bit, write "1" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

<Note>

To prevent the CPU from executing the next instruction before moving to main timer mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_timer, R0     ; TIMER bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to main timer mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

■ Return from the main timer mode

The CPU returns from main timer mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - Main timer interrupt
  - Sub timer interrupt
  - Watch counter interrupt
  - External interrupt
  - An interrupt by the WKUP bit of the USB function

For the interrupt level, see "CHAPTER 10 Interrupt Controller".

<Note>

If the interrupt request is not accepted by the CPU when returning from main timer mode due to the interrupt request, the program is executed starting from the next instruction after entering main timer mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.4.5 Operation in Watch Mode

Watch mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state. The permitted clock oscillation, however, operates, allowing less reduction in power dissipation than in stop mode.
In watch mode, select the sub clock (SBCLK) oscillation as a source clock (SRCCLK) for the CPU.
If watch mode is entered, it continues until a return resource occurs. When a return resource occurs, it returns to the program operation after two or three clock period.

■ Overview

In watch mode, because sub clock (SBCLK) oscillation is permitted as a source clock (SRCCLK) for the CPU, the count operation of the sub timer and watch counter is executed.

■ Setting

Table 8.4-5 shows the settings required before changing to watch mode.

Table 8.4-5 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects sub clock (SBCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=11)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
	MCEN	Stops main clock (MCLK) oscillation (MCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in watch mode 0 = Retain the state in effect before watch mode is entered 1 = Hi-Z

<Note>

When moving to watch mode, if the SLVL1 bit of the standby mode control register (STBCR) is set to "0" while setting doze mode, the DOZE bit is cleared to "0" on returning from watch mode to end doze mode.

■ Transition

By following the steps below, power dissipation mode moves to watch mode.

1. Write "0" to the STOP bit, write "1" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

<Note>

To prevent the CPU from executing the next instruction before moving to watch mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_timer, R0     ; TIMER bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to watch mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

■ Return from the watch mode

The CPU returns from watch mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - Sub timer interrupt request
  - Watch counter interrupt request
  - External interrupt request
  - An interrupt by the WKUP bit of the USB function.

For the interrupt level, see "CHAPTER 10 Interrupt Controller".

<Note>

If the interrupt request is not accepted by the CPU when returning from watch mode due to the interrupt request, the program is executed starting from the next instruction after entering watch mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.4.6 Operation in Stop Mode

Watch mode is categorized as a standby mode. Standby mode stops the entire device to put it in a standby state. By doing so, it can significantly reduce power dissipation in the external event wait state.
Stop mode stops all operations including the oscillation of all clocks to minimize power dissipation.

■ Overview

Using stop mode can minimize power dissipation by stopping the oscillation of all clocks.

To return to the program operation after the return request is generated, however, a certain amount of oscillation stabilization wait time is required.

■ Setting

The setting may differ depending on the source clock of the CPU (SRCCLK) before entering stop mode and after returning from stop mode.

● If the source clock (SRCCLK) of the CPU before/after stop mode is a sub clock (SBCLK)

Table 8.4-6 shows the settings required before changing to stop mode.

Table 8.4-6 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects sub clock (SBCLK) for the CPU source clock (SRCCLK) (CKS1, CKS0=11)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in stop mode 0 = Retain the state in effect before stop mode is entered 1 = Hi-Z

<Note>

At transition to stop mode, if the SLVL1 bit of standby mode control register (STBCR) is set to "0" while doze mode has been set, the DOZE bit is cleared to "0" when the CPU returns from stop mode to end doze mode.

● If the source clock (SRCCLK) of the CPU before/after stop mode is a main clock (MCLK)

Table 8.4-7 shows the settings required before changing to stop mode.

Table 8.4-7 Setting register

Registers	Bit	Explanation
Clock source select register (CSELR)	CKS1, CKS0	Selects the main clock (MCLK) as a source clock (SRCCLK) of the CPU (CKS1, CKS0=00/01)
	PCEN	Stops PLL clock (PLLCLK) oscillation (PCEN = 0)
Standby mode control register (STBCR)	SLVL1	Sets the output signal from the pins in stop mode 0 = Retain the state in effect before stop mode is entered 1 = Hi-Z

<Note>

At transition to stop mode, if the SLVL1 bit of standby mode control register (STBCR) is set to "0" while doze mode has been set, the DOZE bit is cleared to "0" when the CPU returns from stop mode to end doze mode.

■ Transition

By following the steps below, power dissipation mode moves to stop mode.

1. Write "1" to the STOP bit write "0" to the TIMER bit, and write "0" to the SLEEP bit in the standby mode control register (STBCR).
2. Read the standby mode control register (STBCR).

<Note>

To prevent the CPU from executing the next instruction before moving to stop mode, perform the dummy processing that uses the value which is read in the instruction subsequent to step 2, as shown in the example.

Example)

```
LDI     #value_of_stop, R0      ; STOP bit = 1, SLVL1, SLVL0 bit setting
LDI     #_STBCR, R12            ;
STB     R0, @R12                ; write
LDUB    @R12, R0                ; read (move to stop mode)
MOV     R0, R0                  ; dummy processing
NOP                             ; dummy processing
NOP                             ; dummy processing
```

■ Return from stop mode

The CPU returns from stop mode in either of the following cases.

- This device is reset.
- Below interrupt requests are generated (whose interrupt level is other than "31").
  - External interrupt
  - An interrupt by the WKUP bit of the USB function

For information on the interrupt level, see "CHAPTER 10 Interrupt Controller".

<Note>

If the interrupt request is not accepted by the CPU when returning from stop mode due to the interrupt request, the program is executed starting from the next instruction after entering stop mode.If the interrupt request is accepted by the CPU, the operation is branched to the interrupt processing routine.

8.5 Notes on Use

Note the following points on using low-power dissipation mode.

- If the interrupt request is generated when low-power dissipation mode is switched to the following modes, the switching is disabled.
  - Doze mode
  - Sleep mode
  - Main timer mode
  - Watch mode
  - Stop mode
- For instance, sleep mode is not entered in the following cases. Move to sleep mode after clearing the interrupt request.
  - In sleep mode, when returning from sleep mode due to an interrupt request that has not been accepted by the CPU, an operation to move to sleep mode is performed again without clearing the interrupt request.

CHAPTER 9 Reset

This chapter explains the functions and operations of reset.

9.1 Overview
9.2 Configuration
9.3 Pins
9.4 Registers
9.5 Explanation of Operations
9.6 Operating State and Transition

9.1 Overview

This section explains "reset" to initialize the internal circuit.

■ Overview

This device has the following three types of reset resource.

- $\overline{\text{INIT}}$ pin input
- Watchdog reset 0
- Software reset

If either one of the reset resources occurs, operation of all the programs and internal circuits is stopped for initialization.

This state is called a reset state.

If the reset resource is released, operation of the programs and the hardware starts.

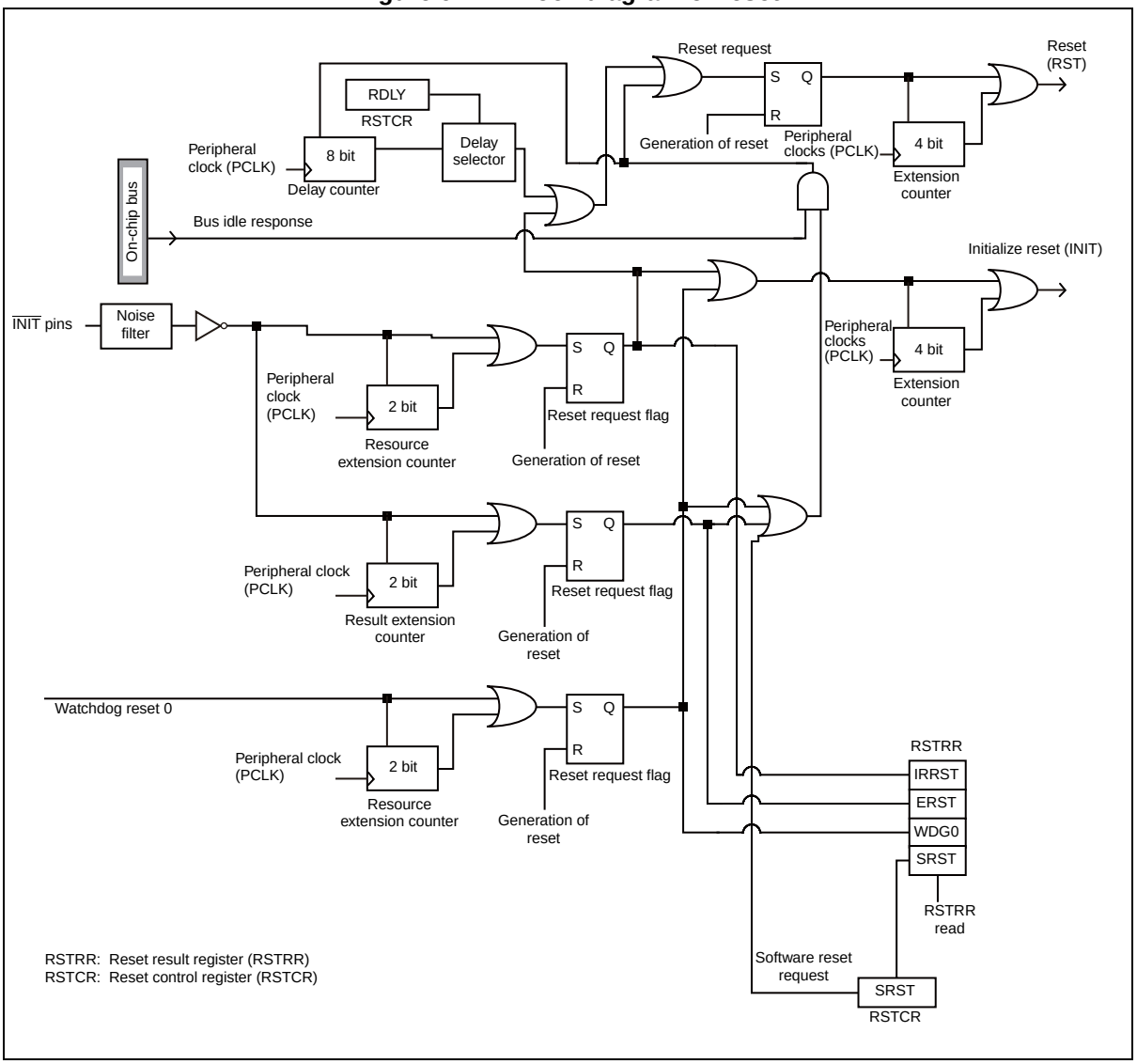
9.2 Configuration

The configuration of reset is shown.

■ Block diagram of reset

Figure 9.2-1 is a block diagram of reset.

Figure 9.2-1 Block diagram of reset



- Reset result register (RSTRR)

  This register indicates the reset resource.
- Reset control register (RSTCR)

  This register controls issuing of reset.
- Delay counter

  This counter counts the period from generation of the reset request until the bus enters the idle state.

  If the bus does not enter the idle state within a certain period of time, the initialize reset (INIT) is forcibly issued.
- Result extension counter

  This counter counts the amount of time for the reset resource to be extended. Each reset resource will be retained until reset is issued.

■ Clocks

Table 9.2-1 shows clocks to be used for reset.

Table 9.2-1 Clocks used for reset

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

9.3 Pins

This section explains the pins that are used for reset.

■ Overview

The following pins are used for reset.

- $\overline{\text{INIT}}$ pins

  The external input pins are used to input the reset request.

9.4 Registers

This section explains the configuration and functions of registers used for reset.

■ List of registers used for reset

Table 9.4-1 shows the list of registers used for reset.

Table 9.4-1 List of registers used for reset

Abbreviated Register Name	Register Name	Reference
RSTRR	Reset result register	9.4.1
RSTCR	Reset control register	9.4.2

9.4.1 Reset Result Register (RSTRR)

This register stores the reset resource.

It stores all the reset resources that have occurred since the power was turned on until this register is read.

Figure 9.4-1 shows the bit configuration of the reset result register (RSTRR).

Figure 9.4-1 Bit configuration of the reset result register (RSTRR)

bit	7	6	5	4	3	2	1	0
	IRRST	ERST	Undefined	WDG0	Undefined	Undefined	Undefined	SRST
Attribute	R	R	R	R	R	R	R	R
Initial value	* This differs depending on the reset resource.							

R: Read only

*: The initial values are as follows:

Reset Resource	Initial Value
$\overline{\text{INIT}}$ pin input	11XXXXXX
Watchdog reset 0	XXX1XXXX
Timeout of the watchdog reset 0	1XX1XXXX
Software reset	XXXXXXX1
Timeout for software reset	1XXXXXX1
Register reading	00000000

<Note>

If this register is read, all the bits are cleared.

[bit7]: IRRST (Irregular reset bit)

A reset is issued without waiting for completion of bus access. This is called an irregular reset. If an irregular reset occurs, the contents of the memory may be damaged.

If either a reset by the $\overline{\text{INIT}}$ pins input or a reset timeout occurs, this bit changes to "1".

Read Value	Explanation
0	No irregular reset is detected. The memory contents are guaranteed to be damage free.
1	An irregular reset is detected. The contents of the memory may have been damaged during the last reset.

For details of the irregular reset, see "■ Irregular reset" in "9.5.3 Operation of Reset".

[bit6]: ERST (Reset pin input bit)

This bits indicates whether the reset by an $\overline{\text{INIT}}$ pin input has occurred.

Read Value	Explanation
0	Reset by an $\overline{\text{INIT}}$ pin input has not occurred.
1	Reset by an $\overline{\text{INIT}}$ pin input has occurred.

[bit5]: Undefined bit

In case of reading	A value is undefined.

[bit4]: WDG0 (Watchdog reset 0 bit)

This bit indicates whether the watchdog reset 0 has occurred.

If a reset timeout occurred in watchdog timer 0, the IRRST bit also changes to "1".

Read Value	Explanation
0	A watchdog reset 0 has not occurred.
1	A watchdog reset 0 has occurred.

[bit3 to bit1]: Undefined bits

In case of reading	A value is undefined.

[bit0]: SRST (Software reset bit)

This bit indicates whether a software reset (RSTCR:SRST) has occurred.

If a reset timeout occurred in the software reset (RSTCR:SRST), the IRRST bit also changes to "1".

Read Value	Explanation
0	A software reset (RSTCR:SRST) has not occurred.
1	A software reset (RSTCR:SRST) has occurred.

9.4.2 Reset Control Register (RSTCR)

This register controls issuing of reset.

Figure 9.4-2 shows the bit configuration of the reset control register (RSTCR).

Figure 9.4-2 Bit configuration of the reset control register (RSTCR)

bit	7	6	5	4	3	2	1	0
	RDLY2	RDLY1	RDLY0	Reserved	Reserved	Reserved	Reserved	SRST
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7 to bit5]: RDLY2 to RDLY0 (Reset issue delay bit)

These bits set the delay time for reset issuing, meaning the length of time that it takes for all the busses to become idle after acceptance of the reset request (delay cycle).

RDLY2	RDLY1	RDLY0	Explanation
0	0	0	Peripheral clock (PCLK) × 2 cycles
0	0	1	Peripheral clock (PCLK) × 4 cycles
0	1	0	Peripheral clock (PCLK) × 8 cycles
0	1	1	Peripheral clock (PCLK) × 16 cycles
1	0	0	Peripheral clock (PCLK) × 32 cycles
1	0	1	Peripheral clock (PCLK) × 64 cycles
1	1	0	Peripheral clock (PCLK) × 128 cycles
1	1	1	Peripheral clock (PCLK) × 256 cycles

<Notes>

- The values of each bit are initialized by reset. Writing after reset is possible only once.
- If a low value is set for the delay cycle, a irregular reset due to the reset timeout will likely occur. In contrast, if a high value is set for the delay cycle, it may take long for the reset to be issued after the reset resource occurs.
- For information on the irregular reset, see "■ Irregular reset" in "9.5.3 Operation of Reset".

[bit4 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: SRST (Software reset bit)

A software reset request occurs if the reset control register (RSTCR) is read after "1" is written to this bit.

Written Value	Explanation
0	A reset request has not occurred.
1	A reset request has occurred by reading this register.

<Notes>

- After "1" is written to this bit, any subsequent writing in the reset control register (RSTCR) is ignored until reset occurs.
- Before generating a software reset request by writing "1" to SRST bit, switch the source clock to the main clock (MCLK) divided by 2.

9.5 Explanation of Operations

This section explains the operation of reset.

9.5.1 Reset Types

Three types of resets are provided for this device, whose reset resources and contents for initialization differ from one another.

- Power-on reset (SINIT)

  This reset is used to initialize the unstable state of the division circuit.

  At the same time, initialize reset (INIT) and reset (RST) are issued.

Reset resource	- Input "L" level to $\overline{\text{INIT}}$ pin
Target of initialization	- Oscillation stabilization wait time of the main clock (MCLK)
Reset that concurrently occurs	- Initialize reset (INIT) - Reset (RST)

- Initialize reset (INIT)

  Initializes the following registers to reset the clock control settings.

  - Clock source select register (CSELR)
  - Clock source monitor register (CMONR)
  - PLL configuration register (PLLCR)
  - Clock stabilization time select register (CSTBR)

  Reset (RST) is issued at the same time.

Reset resource	- $\overline{\text{INIT}}$ pin input - Reset time out - Watchdog reset 0
Target of initialization	- Source clock = Main clock (MCLK) divided by 2 - Clock oscillation = Main clock oscillates, sub/PLL clock stopped - Division rate of the PLL macro oscillation clock - Multiplying factor of the PLL clock (PLLCLK) - Oscillation stabilization wait time of the PLL clock - Division rate of the PLL input clock - Oscillation stabilization wait time of the sub clock
Reset that concurrently occurs	- Reset (RST)

- Reset (RST)

  This reset initializes the program operation.

Reset resource	- $\overline{\text{INIT}}$ pin input - Reset time out - Watchdog reset 0 - Software reset
Target of initialization	All the register settings and hardware other than those that are initialized by the power-on reset (SINIT) and initialize reset (INIT).
Reset that concurrently occurs	No

9.5.2 Reset Resource

There are three types of reset resource. The level of the reset that is issued differs depending on the reset resource.

In addition, whether there is an occurrence of the irregular reset that issues initialize reset (INIT) without verifying completion of bus access, also depends on the reset resource.

- $\overline{\text{INIT}}$ pin input

  An initialize reset (INIT) request occurs while "L" level is input in the $\overline{\text{INIT}}$ pin.

Generation source	"L" level is input in the $\overline{\text{INIT}}$ pin
Cancellation source	"H" level is input in the $\overline{\text{INIT}}$ pin
Reset level	Issues all of the three resets: power-on reset (SINIT), initialize reset (INIT), and reset (RST)
Corresponding flag	ERST bit of the reset result register (RSTRR) = 1
Operation	Issues the power-on reset (SINIT), initialize reset (INIT), and reset (RST) without waiting for a completion of bus access (irregular reset).

- Watchdog reset 0

  The watchdog reset 0 request is generated if the period set for the watchdog timer elapses. If the watchdog reset 0 request is generated, the initialize reset (INIT) is issued.

Generation source	The period set for the watchdog timer elapses
Cancellation source	Automatically cancelled after the initialize reset (INIT) is issued.
Reset level	Issues the initialize reset (INIT) and reset (RST)
Corresponding flag	WDG0 bit of the reset result register (RSTRR) = 1
Operation	- Issues an initialize reset (INIT) and reset (RST) after the completion of bus access is verified. - Forcibly issues an initialize reset (INIT) and reset (RST) if a reset timeout occurs before completion of bus access (irregular reset).

- Software reset (RSTCR:SRST)

  If the reset control resister (RSTCR) is read after "1" is written to the SRST bit of the reset control register (RSTCR), a reset (RST) request is generated.

Generation source	The reset control register (RSTCR) is read after "1" is written to the SRST bit of the reset control register (RSTCR). * Set the main clock (MCLK) to the source clock (SRCCLK) before writing "1" to SRST bit.
Cancellation source	Automatically cancelled after the reset (RST) is issued.
Reset level	Issues only reset (RST)
Corresponding flag	SRST bit of the reset result register (RSTRR) = 1
Operation	- Issues reset (RST) after verifying completion of bus access. - Forcibly issues an initialize reset (INIT) and reset (RST) if a reset timeout occurs before completion of bus access (irregular reset).

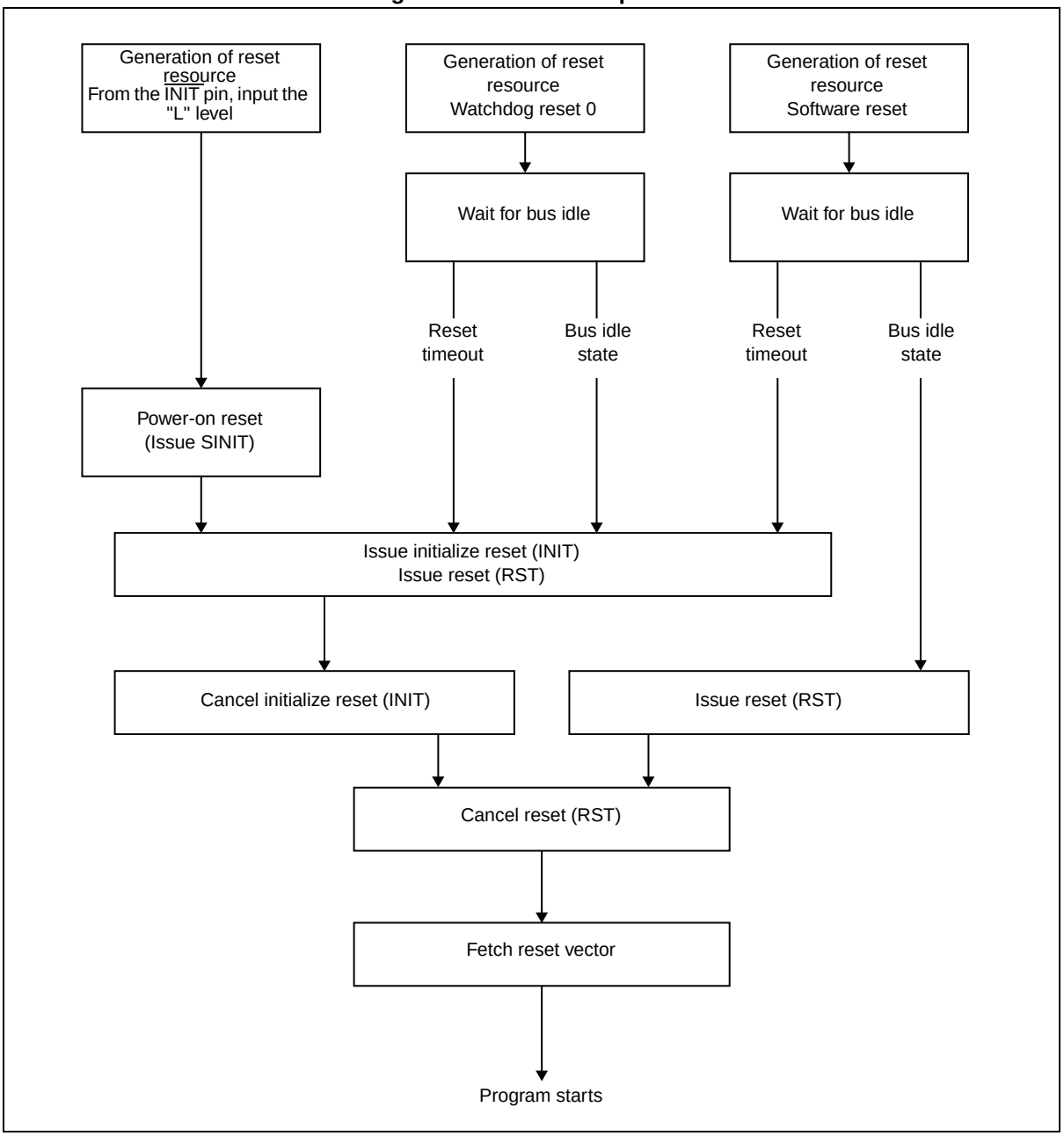
9.5.3 Operation of Reset

■ Flow of reset operation

A series of operations from the generation of reset, through reset state, until the CPU starts operation is called a reset sequence.

Figure 9.5-1 shows the reset sequence.

Figure 9.5-1 Reset sequence


Generation of reset resource From the INIT pin, input the "L" level
Generation of reset resource Watchdog reset 0
Generation of reset resource Software reset
Wait for bus idle
Wait for bus idle
Reset timeout
Bus idle state
Reset timeout
Bus idle state
Power-on reset (Issue SINIT)
Issue initialize reset (INIT) Issue reset (RST)
Cancel initialize reset (INIT)
Issue reset (RST)
Cancel reset (RST)
Fetch reset vector
Program starts

1. Retrieval and extension of reset resource

   The generated reset resource is asynchronously retrieved and retained until reset is issued.

   2 bits of resource extension counter retains the reset resource for at least 4Ts (T: Peripheral clock (PCLK) period).

2. Generation of the reset request

   Reports the generated reset request to the internal bus controller to perform the following processing.

   - Stops the program operation of the CPU (same as for sleep mode).
   - Verifies that the idle request has been reported to all busses.

   At the same time, the delay counter starts counting.

3. Acceptance of reset request and issue of reset

   After all processing for the reset request is completed, the reset request is accepted.

   An irregular reset is issued if a reset timeout occurs due to an overflow of the delay counter before response of the completion from the bus.

4. Issue of reset
   - Input "L" level to $\overline{\text{INIT}}$ pin

     Issues a power-on reset (SINIT), initialize reset (INIT), and reset (RST).
   - Watchdog reset 0

     Issues initialize reset (INIT) and reset (RST).
   - Reset time out

     Issues initialize reset (INIT) and reset (RST).
   - Software reset (RSTCR:SRST)

     Issues reset (RST).

5. Cancellation of reset resource

   If the reset resource is cancelled, the reset request is extended for a period of 4Ts (T: Peripheral clock (PCLK)). The request is then retained for 16 Ts (T: Peripheral clock (PCLK)) reset period. Therefore, the minimum cycle of reset issue is 20 Ts.

6. Cancellation of reset

   When the reset cycle ends, reset is cancelled and the hardware starts operation.

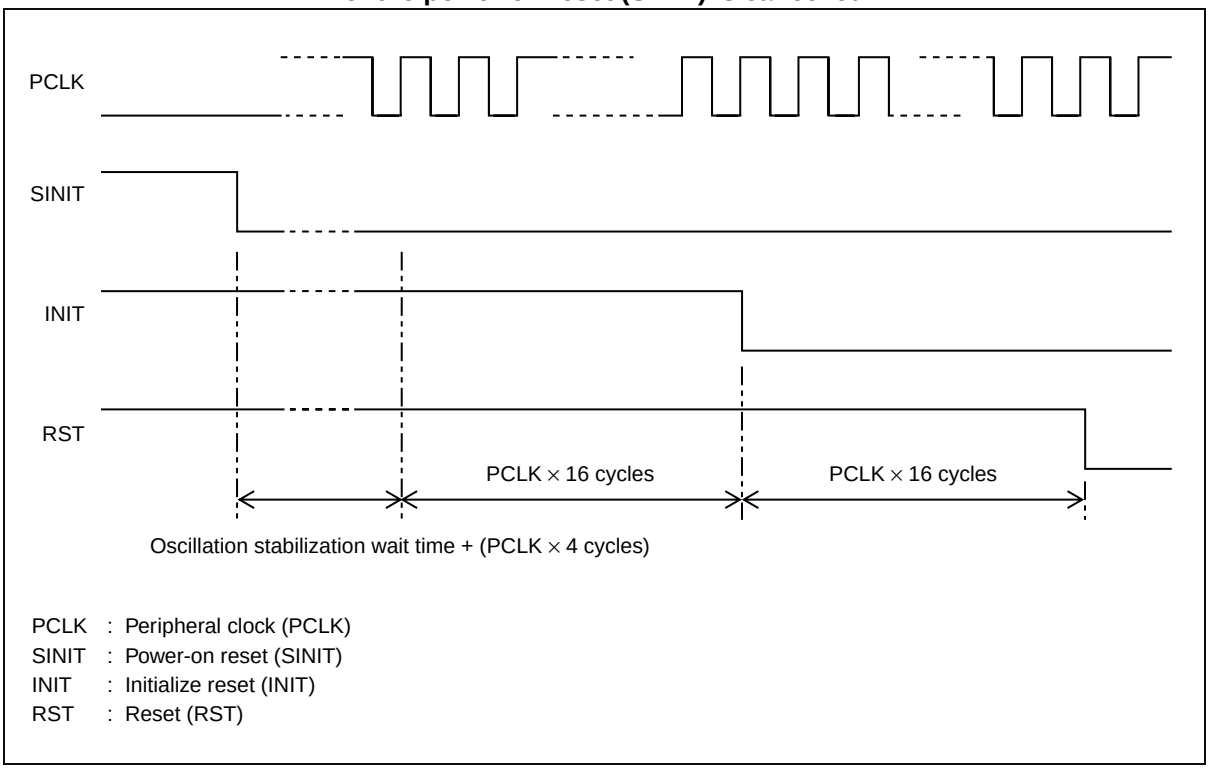
7. Retrieval of the reset vector (fetch)

   The CPU starts fetching the reset vector (000F $FFFC_H$). The CPU retrieves the fetched reset vector in the program counter (PC) to start program operation.

■ Power-on reset (SINIT)

Initialize reset (INIT) and reset (RST) are also issued at the same time as the power-on reset (SINIT) is issued. Figure 9.5-2 shows the respective reset issue sequence after the reset resource of the power-on reset (SINIT) is cancelled.

Figure 9.5-2 Each reset issue sequence after the reset resource of the power-on reset (SINIT) is cancelled

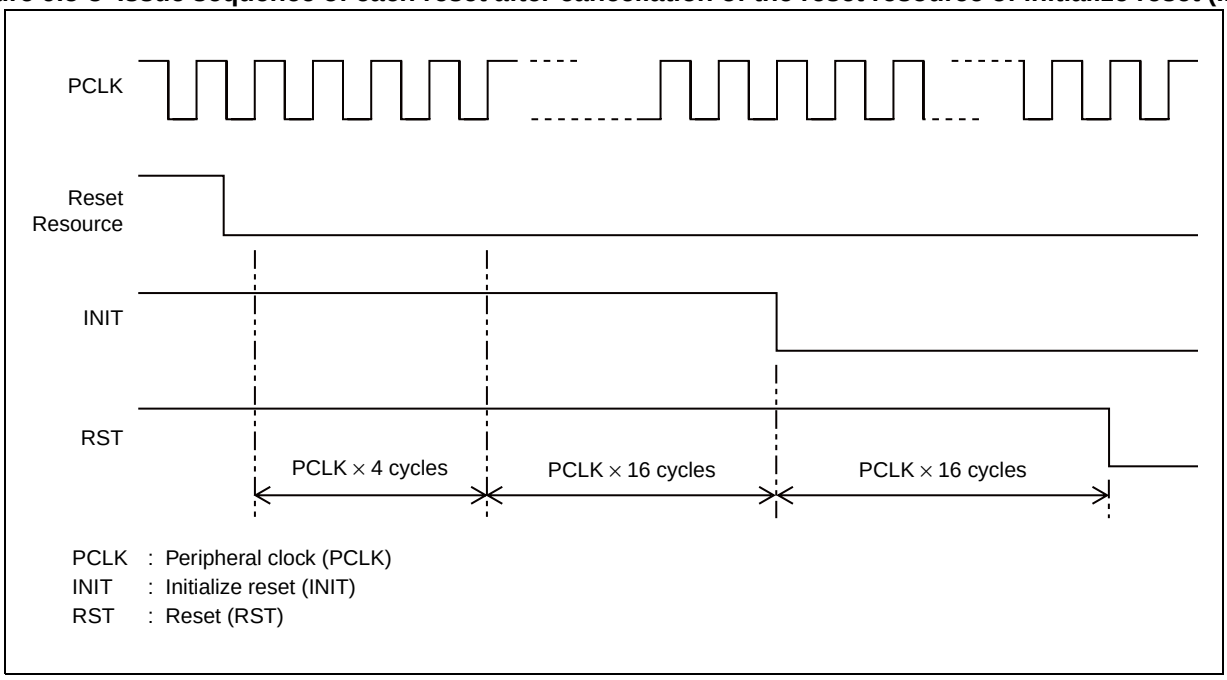


■ Initialize reset (INIT)

When initialize reset (INIT) is issued, reset (RST) is also issued at the same time.

Figure 9.5-3 shows the issue sequence of the respective resets after the reset resource of initialize reset (INIT) is cancelled.

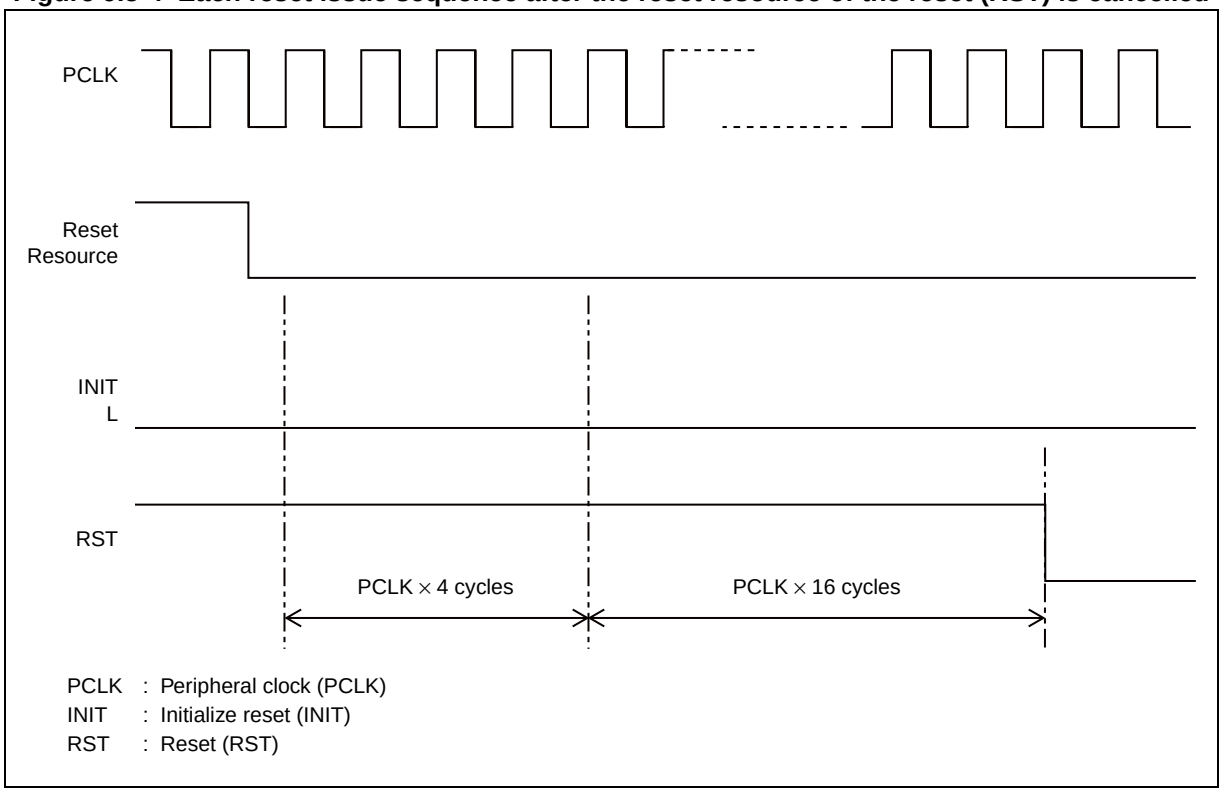
Figure 9.5-3 Issue sequence of each reset after cancellation of the reset resource of initialize reset (INIT)



■ Reset (RST)

Figure 9.5-4 shows the respective reset issue sequence after the reset resource of reset (RST) is cancelled.

Figure 9.5-4 Each reset issue sequence after the reset resource of the reset (RST) is cancelled



■ Irregular reset

Irregular reset occurs in the following cases.

- When an $\overline{\text{INIT}}$ pin input (INIT) is used
- When a reset timeout occurs

  (The delay counter overflows before the response from the bus is received during watchdog reset 0 / software reset (RSTCR: SRST).)

If irregular reset occurs, the following processes are executed.

- Initialize reset (INIT) is issued.
- The IRRST bit of the reset result register (RSTRR) changes to "1".

<Note>

When irregular reset occurs, the bus access may be performed at the time of reset input. In this case, the contents of the memory may be damaged.

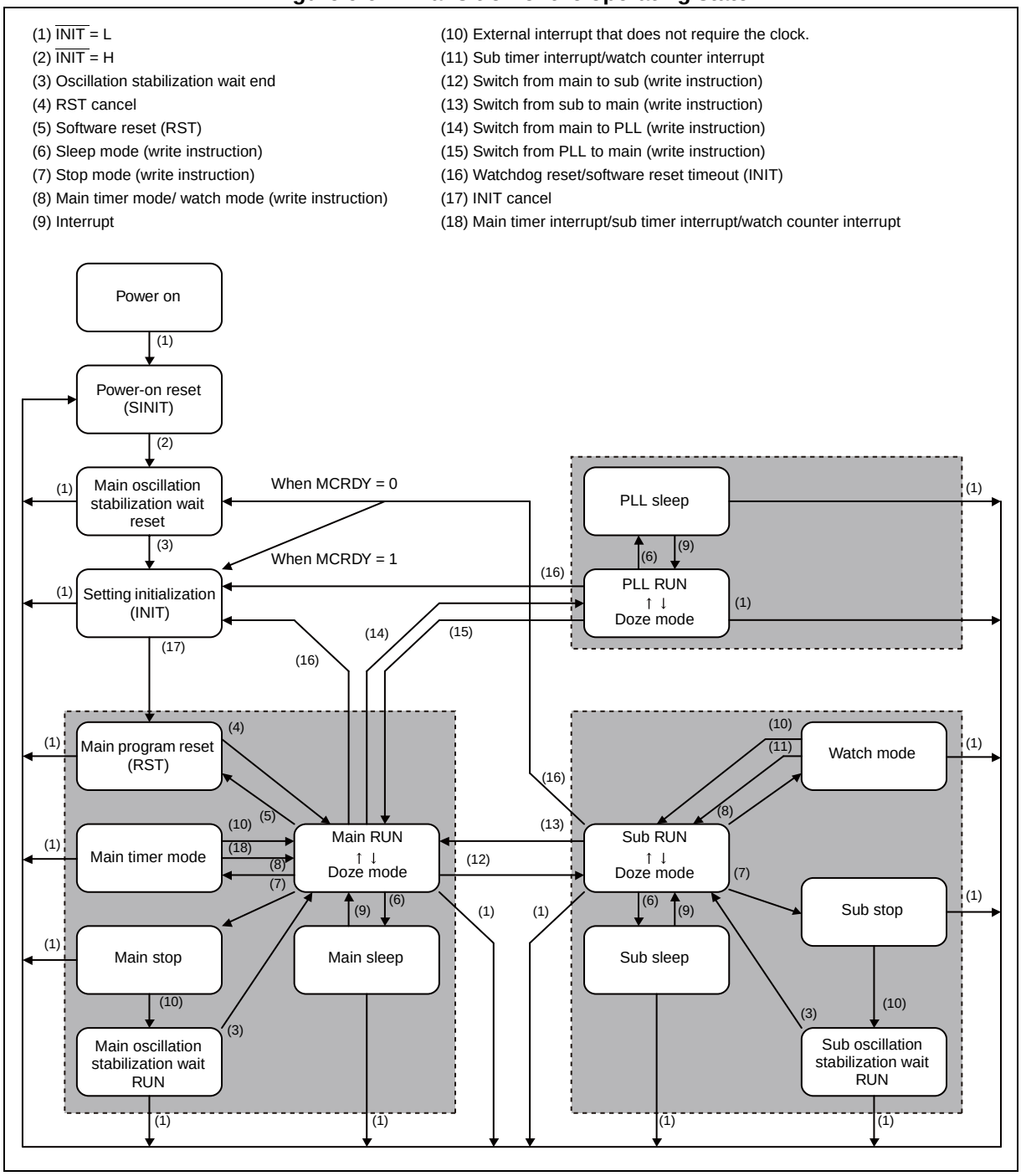
9.6 Operating State and Transition

This section explains each operating state and how to control it.

■ Operating state

Figure 9.6-1 shows transition of the operating state.

Figure 9.6-1 Transition of the operating state


(1) INIT = L
(2) INIT = H
(3) Oscillation stabilization wait end
(4) RST cancel
(5) Software reset (RST)
(6) Sleep mode (write instruction)
(7) Stop mode (write instruction)
(8) Main timer mode/ watch mode (write instruction)
(9) Interrupt
(10) External interrupt that does not require the clock.
(11) Sub timer interrupt/watch counter interrupt
(12) Switch from main to sub (write instruction)
(13) Switch from sub to main (write instruction)
(14) Switch from main to PLL (write instruction)
(15) Switch from PLL to main (write instruction)
(16) Watchdog reset/software reset timeout (INIT)
(17) INIT cancel
(18) Main timer interrupt/sub timer interrupt/watch counter interrupt
Power on
Power-on reset (SINIT)
Main oscillation stabilization wait reset
When MCRDY = 0
When MCRDY = 1
Setting initialization (INIT)
PLL sleep
PLL RUN ↑ ↓ Doze mode
Main program reset (RST)
Main timer mode
Main stop
Main oscillation stabilization wait RUN
Main RUN ↑ ↓ Doze mode
Main sleep
Sub RUN ↑ ↓ Doze mode
Sub sleep
Watch mode
Sub stop
Sub oscillation stabilization wait RUN

● RUN state (normal operation)

Program is running.

All the internal clocks are delivered and all the circuits are enabled.

The Hi-Z control of the external pins in stop state, main timer mode state and watch mode state is cancelled.

● Sleep state

Program is stopped. Transition occurs by program operation.

Only program execution of the CPU is stopped. The peripheral circuits are enabled.

The built-in memories and external busses are suspended until the DMA controller (DMAC) request is received.

In bus sleep mode, the internal bus is suspended until the DMA controller (DMAC) request is received.

- If a valid interrupt request is generated, the device undergoes transition to the RUN state (normal operation).
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

● Watch mode state

The device is in a suspended state. Transition occurs by the program operation.

Internal circuits other than the oscillation circuits (sub clock (SBCLK)) are stopped.

The external pins can be uniformly set to Hi-Z (excluding certain pins).

- If an external interrupt request is generated, the device undergoes transition to the RUN state (normal operation).
- If a sub timer interrupt, or watch counter interrupt request is generated, it undergoes transition to the RUN state (normal operation).
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

<Note>

Stop oscillation of the main clock (MCLK) and PLL clock (PLLCLK) before transition to watch mode.

● Main timer mode state

The device is in a suspended state. Transition occurs by the program operation.

Internal circuits other than the oscillation circuits (main clock (MCLK) and sub clock (SBCLK)) are stopped.

The external pins can be uniformly set to Hi-Z (excluding certain pins).

- If an external interrupt is generated, the device undergoes transition to the RUN state (normal operation).
- If a main timer interrupt, sub timer interrupt, and watch counter interrupt requests are generated, it undergoes transition to the RUN state (normal operation).
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

<Note>

Stop oscillation of the PLL clock (PLLCLK) before transition to main timer mode.

● Stop state

The device is in a suspended state. Transition occurs by the program operation.

All the internal circuits are suspended.

The external pins can be uniformly set to Hi-Z (excluding certain pins).

- If an external interrupt request is generated, the device undergoes transition to the oscillation stabilization wait RUN state.
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

<Note>

Stop oscillation of the PLL clock (PLLCLK) before transition to the stop state.

● Oscillation stabilization wait RUN state

The device is in a suspended state. Transition to this state occurs after the device returns from the stop state.

All the internal circuits are suspended (excluding timer operation for clock stabilization wait).

While all the internal clocks are stopped, oscillation circuits that have been enabled operate.

- When the oscillation stabilization wait time elapses, the device undergoes transition to the RUN state (normal operation).
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

● Oscillation stabilization wait reset (RST) state

The device is in a suspended state. Transition occurs after the device returns from power-on reset (SINIT).

All the internal circuits are suspended (excluding timer operation for oscillation stabilization wait).

While all the internal clocks are suspended, the main oscillation circuit operates.

- When the oscillation stabilization wait time elapses, the device undergoes transition to the initialize reset (INIT) state.
- If "L" level is input in the $\overline{\text{INIT}}$ pin, it undergoes transition to the power-on reset (SINIT) state.

● Program reset (RST) state

Program is in the initialized state. Transition occurs when a reset (RST) request is accepted or after the initialize reset (INIT) state ends.

The program execution of the CPU is suspended and the program counter is initialized. The peripheral circuits are initialized (excluding certain circuits).

All the internal clocks as well as the oscillation circuits that have been enabled and the PLL clock (PLLCLK) operate.

- The reset (RST) request for the internal circuits is generated. When the reset (RST) request disappears, transition to the RUN state (normal operation) occurs.
- If "L" level is input in the $\overline{\text{INIT}}$ pin, the device undergoes transition to the power-on reset (SINIT) state.

● Initialize reset (INIT) state

This is the state in which all settings are initialized. Transition occurs when the initialize reset (INIT) request is accepted.

The program execution of the CPU is suspended and the program counter is initialized. All the peripheral circuits are initialized. The main clock (MCLK) oscillation circuit operates (while the sub clock (SBCLK) oscillation circuit and PLL clock (PLLCLK) oscillation circuit stop operation). All the internal clocks stop while the "L" level is being input in the $\overline{\text{INIT}}$ pin. Otherwise, they operate.

Initialize reset (INIT) and reset (RST) are output to the internal circuit.

- When the initialize reset (INIT) request disappears, this state is cancelled and transition to the program reset (RST) state occurs.
- If "L" is input in the $\overline{\text{INIT}}$ pin, the device undergoes transition to the power-on reset (SINIT) state.

■ Priority of state transition requests

state transition requests are prioritized in the following order in any states. However, since some requests are generated only in the particular states, they are enabled only in those states.

Highest priority	Power-on reset (SINIT) request	
	Initialize reset (INIT) request	
	Oscillation stabilization wait time end	Occurs only in the oscillation stabilization wait reset state and oscillation stabilization wait RUN state
	Reset (RST) request	
	Valid interrupt request	Occurs only in the RUN, sleep, stop, and watch mode state
	Stop mode request (register write)	Occurs only in the RUN state
	Watch mode request (register write)	Occurs only in the RUN state
Lowest priority	Sleep mode request (register write)	Occurs only in the RUN state

CHAPTER 10 Interrupt Controller

This chapter explains the functions and operations of the interrupt controller.

10.1 Overview

10.2 Configuration

10.3 Registers

10.4 An Explanation of Operations and Setting Procedure Examples

10.5 Notes on Use

10.1 Overview

The interrupt controller determines the priority of an interrupt request and sends the request to the CPU.

■ Overview

The interrupt control has the following functions:

- Accepts interrupt requests from peripheral functions.
- Determines the priority of sending interrupt requests to the CPU according to the interrupt level and interrupt vector.
- Sends the highest priority interrupt request to the CPU.
- Sends the interrupt vector number of the highest priority interrupt request to the CPU.
- Generates a request for returning from sleep mode or stop mode according to an interrupt request with an interrupt level other than "1111".

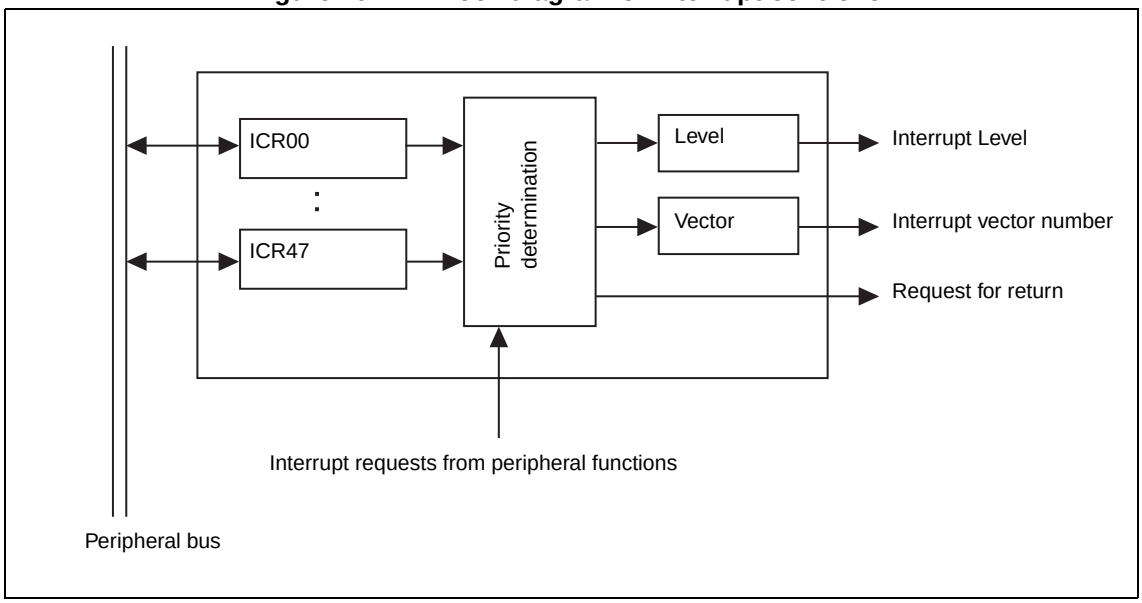
10.2 Configuration

This section explains the interrupt controller configuration.

■ Block diagram of interrupt controller

Figure 10.2-1 shows a block diagram of the interrupt controller.

Figure 10.2-1 Block diagram of interrupt controller



- Interrupt priority determination circuit

  This circuit determines the priority of an incoming interrupt request. It also generates a request to return from sleep mode or stop mode.
- Interrupt level generating circuit

  This circuit transmits the interrupt level of an interrupt request to the CPU.
- Interrupt vector generating circuit

  This circuit sends the interrupt vector of an interrupt request to the CPU.
- Interrupt control registers (ICR00 to ICR47)

  These registers are used to set the interrupt levels of interrupt requests.

■ Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

10.3 Registers

This section explains the configurations and functions of the registers used by the interrupt controller.

■ Interrupt controller registers

Table 10.3-1 lists the interrupt controller registers.

Table 10.3-1 Interrupt controller registers

Abbreviated Register Name	Register Name	Reference
ICR00 to ICR47	Interrupt control registers 00 to 47	10.3.1

10.3.1 Interrupt Control Register (ICR00 to ICR47)

These registers are used to set interrupt levels. This register is provided for input of each interrupt.

Figure 10.3-1 shows the bit configuration of the interrupt control registers (ICR00 to ICR47).

Figure 10.3-1 Bit configuration of interrupt control registers (ICR00 to ICR47)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	IL4	IL3	IL2	IL1	IL0
Attribute	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	1

R/W: Read/Write

R: Read only

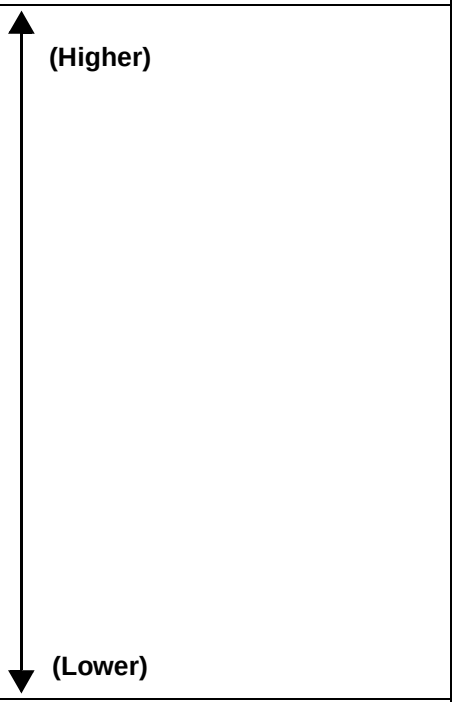
[bit7 to bit5]: Undefined bits

In case of writing	Ignored
In case of reading	"1" is read.

[bit4 to bit0]: IL4 to IL0 (interrupt level control bits)

These bits specify the interrupt level of an interrupt request.

When reset, the bits are initialized to IL4 to IL0=11111("11111_B" is level 31 interrupt disabled).

IL4	IL3	IL2	IL1	IL0	Interrupt Level	
1	0	0	0	0	16	Highest level that can be set
1	0	0	0	1	17	
1	0	0	1	0	18	
1	0	0	1	1	19	
1	0	1	0	0	20	
1	0	1	0	1	21	
1	0	1	1	0	22	
1	0	1	1	1	23	
1	1	0	0	0	24	
1	1	0	0	1	25	
1	1	0	1	0	26	
1	1	0	1	1	27	
1	1	1	0	0	28	
1	1	1	0	1	29	
1	1	1	1	0	30	Lowest level that can be set
1	1	1	1	1	31	Interrupt Disabled

<Notes>

- If the interrupt level that is set in this register is lower than the mask level in the CPU interrupt level mask register (ILM), the interrupt request is masked on the CPU side.
- The interrupt control register (ICR00 to ICR47) in which an interrupt level is set varies depending on the peripheral function. For information on the correspondence between the peripheral function and interrupt control register (ICR00 to ICR47), see "APPENDIX C Interrupt Vectors".
- IL4 bit is fixed to "1" and IL3 to IL0 can be set.

10.4 An Explanation of Operations and Setting Procedure Examples

This section explains the operations of the interrupt controller.

10.4.1 Explanation of Operations of Interrupt Controller

This section explains the three types of operations of the interrupt controller.

- Specifying interrupt levels using interrupt control registers (ICR00 to ICR47)
- Determining the priorities of interrupt requests
- Generating a request to return from sleep mode or stop mode

■ Specifying an interrupt level

The procedure for setting interrupt levels using interrupt control registers (ICR00 to ICR47) is shown below:

1. Set an interrupt level in the interrupt control register (ICR00 to ICR47) with the interrupt vector number corresponding to the peripheral function for which an interrupt request needs to be generated.
   For information on the correspondence between interrupt control numbers and interrupt requests, see "APPENDIX C Interrupt Vectors".
2. Enable generation of interrupt requests on the peripheral function for which an interrupt request needs to be generated.
3. Activate the relevant peripheral function.

■ Determining the priorities of interrupt requests

The interrupt controller sends the interrupt level and interrupt vector number of the highest priority interrupt request, among the interrupt requests that are concurrently generated, to the CPU.

The criteria for determining the priorities of interrupt requests are shown in order of determining:

1. Is the interrupt level of the interrupt request "30" or lower (Level 31 is "Interrupt Disabled").
2. Is the value of the interrupt level of the interrupt request the smallest.
3. If the interrupt level is the same, is the interrupt vector number of the interrupt request the smallest.

If no interrupt request meets the above criteria, interrupt level "31" (11111_B) that indicates no interrupt request is output to the CPU.

■ Generating a request to return from sleep mode

If an interrupt request with an interrupt level other than "31" is generated, the interrupt controller generates a request to the clock control part to return from sleep mode.

■ Generating a request to return from stop mode

If an external interrupt request /USB function with an interrupt level other than "31" is generated, the interrupt controller generates a request to the clock control part to return from stop mode.

After return from the stop mode, the interrupt priority determination circuit resumes operation only after the operation of clock begins. The CPU thus executes instructions until the interrupt priority determination circuit produces results.

<Note>

For interrupts that are not used as causes of return from stop mode, set interrupt level "31" (Interrupt Disabled) in the corresponding interrupt control registers (ICR00 to ICR47).

10.5 Notes on Use

Note the following points about using the interrupt controller.

■ Note on the program

- For interrupt requests that should not be used to generate a request to return from sleep mode or stop mode, set interrupt level "31" (Interrupt Disabled) in the corresponding interrupt control registers (ICR00 to ICR47).

■ Notes on operations

- If the interrupt level that is set in an interrupt control register (ICR00 to ICR47) is lower than the mask level in the CPU interrupt level mask register (ILM), the interrupt request is masked on the CPU side.

CHAPTER 11 Interrupt Request Batch-Read Function

This section explains the interrupt request batch-read function.

11.1 Overview
11.2 Configuration
11.3 Registers
11.4 Notes on Use

11.1 Overview

The interrupt request batch-read function reads multiple interrupt requests assigned to one interrupt vector all at once.
The bit search instruction of an FR80 family CPUs can be used to quickly check which interrupt requests have been generated.

This function allows the user to check at one time whether interrupt requests that use the same interrupt vector number have been generated.

Note that this function cannot clear the interrupt request flag. Use the register of each peripheral function to clear the interrupt request flag.

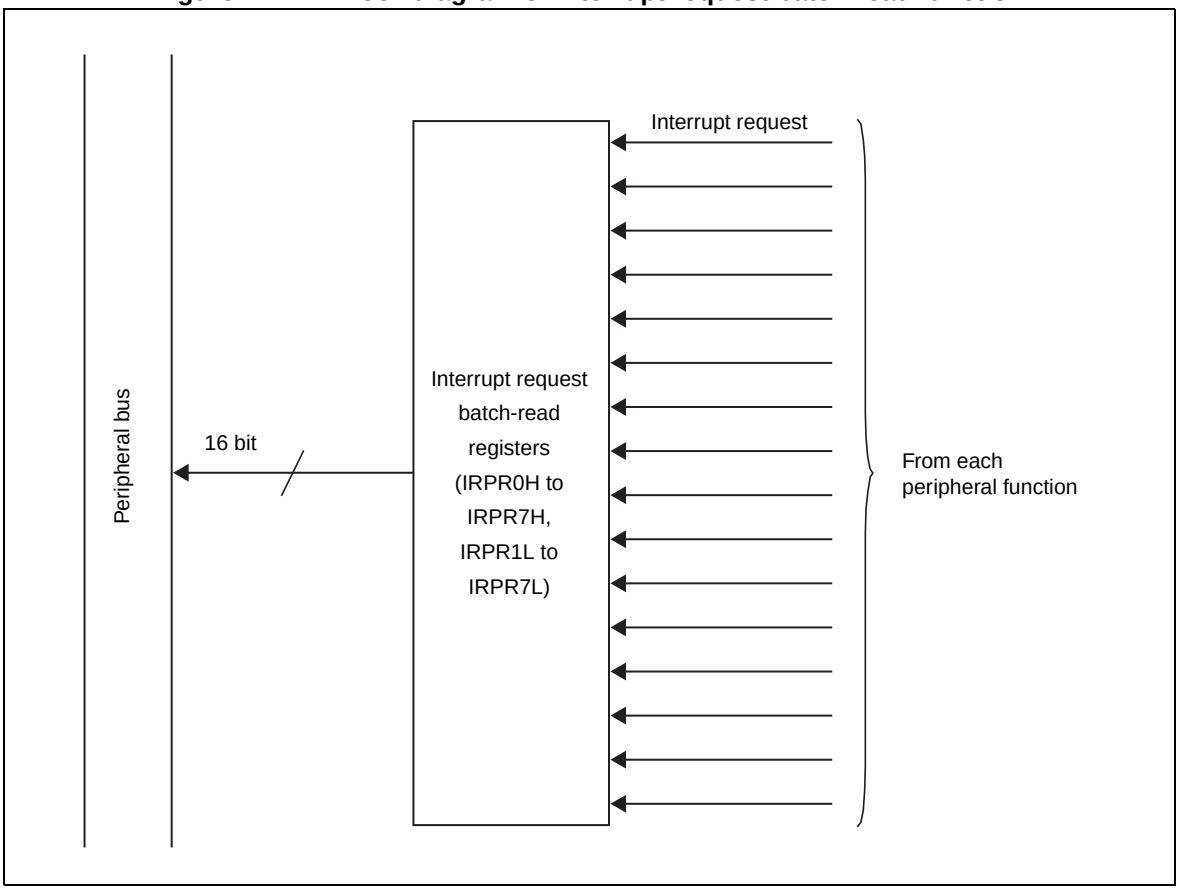
11.2 Configuration

This section shows the configuration of the interrupt request batch-read function.

■ Block diagram of interrupt request batch-read function

Figure 11.2-1 is a block diagram of the interrupt request batch-read function.

Figure 11.2-1 Block diagram of interrupt request batch-read function


Peripheral bus
16 bit
Interrupt request batch-read registers (IRPR0H to IRPR7H, IRPR1L to IRPR7L)
Interrupt request
From each peripheral function

■ Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

11.3 Registers

This section explains the configuration and functions of registers used by the interrupt request batch-read function.

■ Registers for interrupt request batch-read function

Table 11.3-1 lists the registers for the interrupt request batch-read function.

Table 11.3-1 Registers for the interrupt request batch-read function

Abbreviated Register Name	Register Name	Reference
IRPR0H	Interrupt request batch-read register 0 upper	11.3.1
IRPR1H/ IRPR1L	Interrupt request batch-read register 1 upper/lower	11.3.2
IRPR2H/ IRPR2L	Interrupt request batch-read register 2 upper/lower	11.3.3, 11.3.4
IRPR3H/ IRPR3L	Interrupt request batch-read register 3 upper/lower	11.3.5, 11.3.6
IRPR4H/ IRPR4L	Interrupt request batch-read register 4 upper/lower	11.3.7, 11.3.8
IRPR5H/ IRPR5L	Interrupt request batch-read register 5 upper/lower	11.3.9, 11.3.10
IRPR6H/ IRPR6L	Interrupt request batch-read register 6 upper/lower	11.3.11, 11.3.12
IRPR7H/ IRPR7L	Interrupt request batch-read register 7 upper/lower	11.3.13, 11.3.14

11.3.1 Interrupt Request Batch-Read Register 0 Upper (IRPR0H)

The interrupt requests of 16-bit reload timer ch.0 to ch.2 are assigned to interrupt vector number 20 (decimal). This register can be read to check the channel on which an interrupt request has been generated.

Figure 11.3-1 shows the bit configuration of interrupt request batch-read register 0 upper (IRPR0H).

Figure 11.3-1 Bit configuration of interrupt request batch-read register 0 upper (IRPR0H)

Interrupt request batch-read register 0 upper (IRPR0H)

bit	15	14	13	12	11	10	9	8
	RTIR0	RTIR1	RTIR2	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

The bit corresponding to the channel on which an interrupt request has been generated is set to "1".

Bit number	Bit	Value	Explanation
bit15	RTIR0	0	No interrupt request in reload timer ch.0
		1	Interrupt request in reload timer ch.0
bit14	RITR1	0	No interrupt request in reload timer ch.1
		1	Interrupt request in reload timer ch.1
bit13	RTIR2	0	No interrupt request in reload timer ch.2
		1	Interrupt request in reload timer ch.2
bit12 to bit0	Undefined	"0" is read.	

11.3.2 Interrupt Request Batch-Read Register 1 Upper/Lower (IRPR1H/ IRPR1L)

Interrupt vector number 39 (decimal) is used for multifunction serial interface channels ch.8 to ch.11. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-2 shows the bit configuration of interrupt request batch-read register 1 upper/lower (IRPR1H/ IRPR1L).

Figure 11.3-2 Bit configuration of interrupt request batch-read register 1 upper/lower (IRPR1H/ IRPR1L)

Interrupt request batch-read register 1 upper (IRPR1H)

bit	15	14	13	12	11	10	9	8
	RXIR8	TXIR8	ISIR8	Undefined	RXIR9	TXIR9	ISIR9	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Interrupt request batch-read register 1 lower (IRPR1L)

bit	7	6	5	4	3	2	1	0
	RXIR10	TXIR10	ISIR10	Undefined	RXIR11	TXIR11	ISIR11	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	RXIR8	0	No UART/CSIO/I^{2}C receive interrupt request on ch.8
		1	UART/CSIO/I^{2}C receive interrupt request on ch.8
bit14	TXIR8	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.8
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.8
bit13	ISIR8	0	No I^{2}C status interrupt request on ch.8
		1	I^{2}C status interrupt request on ch.8
bit12	Undefined	"0" is read.	
bit11	RXIR9	0	No UART/CSIO/I^{2}C receive interrupt request on ch.9
		1	UART/CSIO/I^{2}C receive interrupt request on ch.9
bit10	TXIR9	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.9
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.9
bit9	ISIR9	0	No I^{2}C status interrupt request on ch.9
		1	I^{2}C status interrupt request on ch.9
bit8	Undefined	"0" is read.	
bit7	RXIR10	0	No UART/CSIO/I^{2}C receive interrupt request on ch.10
		1	UART/CSIO/I^{2}C receive interrupt request on ch.10
bit6	TXIR10	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.10
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.10
bit5	ISIR10	0	No I^{2}C status interrupt request on ch.10
		1	I^{2}C status interrupt request on ch.10
bit4	Undefined	"0" is read.	
bit3	RXIR11	0	No UART/CSIO/I^{2}C receive interrupt request on ch.11
		1	UART/CSIO/I^{2}C receive interrupt request on ch.11

Bit number	Bit	Value	Explanation
bit2	TXIR11	0	No UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.11
		1	UART/CSIO/I^{2}C transmit/transmit bus idle/transmit FIFO interrupt request on ch.11
bit1	ISIR11	0	No I^{2}C status interrupt request on ch.11
		1	I^{2}C status interrupt request on ch.11
bit0	Undefined	"0" is read.	

11.3.3 Interrupt Request Batch-Read Register 2 Upper (IRPR2H)

Interrupt vector number 40 (decimal) is used for 16-bit up/down counter channels ch.0 to ch.3. This register can be read to check the channel on which an interrupt request has been generated.

Figure 11.3-3 shows the bit configuration of interrupt request batch-read register 2 upper (IRPR2H).

Figure 11.3-3 Bit configuration of Interrupt request batch-read register 2 upper (IRPR2H)

bit	15	14	13	12	11	10	9	8
	UDIR0	UDIR1	UDIR2	UDIR3	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	UDIR0	0	No interrupt request in 16-bit up/down counter ch.0
		1	Interrupt request in 16-bit up/down counter ch.0
bit14	UDIR1	0	No interrupt request in 16-bit up/down counter ch.1
		1	Interrupt request in 16-bit up/down counter ch.1
bit13	UDIR2	0	No interrupt request in 16-bit up/down counter ch.2
		1	Interrupt request in 16-bit up/down counter ch.2
bit12	UDIR3	0	No interrupt request in 16-bit up/down counter ch.3
		1	Interrupt request in 16-bit up/down counter ch.3
bit11 to bit8	Undefined	"0" is read.	

11.3.4 Interrupt Request Batch-Read Register 2 Lower (IRPR2L)

Interrupt vector number 41 (decimal) is used for the following peripheral functions:

- Main timer
- Sub timer
- Watch counter

This register can be read to check the peripheral function from which an interrupt request has been generated.

Figure 11.3-4 shows the bit configuration of interrupt request batch-read register 2 lower (IRPR2L).

Figure 11.3-4 Bit configuration of interrupt request batch-read register 2 lower (IRPR2L)

bit	7	6	5	4	3	2	1	0
	MCIR	SCIR	TCIR	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	MCIR	0	No main timer interrupt request
		1	Main timer interrupt request
bit6	SCIR	0	No sub timer interrupt request
		1	Sub timer interrupt request
bit5	TCIR	0	No watch counter interrupt request
		1	Watch counter interrupt request
bit4 to bit0	Undefined	"0" is read.	

11.3.5 Interrupt Request Batch-Read Register 3 Upper (IRPR3H)

Interrupt vector number 44 (decimal) is used for 32-bit input capture channels ch.0 to ch.3. This register can be read to check on which channel an interrupt request has been generated.

Figure 11.3-5 shows the bit configuration of interrupt request batch-read register 3 upper (IRPR3H).

Figure 11.3-5 Bit configuration of Interrupt request batch-read register 3 upper (IRPR3H)

bit	15	14	13	12	11	10	9	8
	ICIR0	ICIR1	ICIR2	ICIR3	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	ICIR0	0	No interrupt request on 32-bit input capture ch.0
		1	Interrupt request on 32-bit input capture ch.0
bit14	ICIR1	0	No interrupt request on 32-bit input capture ch.1
		1	Interrupt request on 32-bit input capture ch.1
bit13	ICIR2	0	No interrupt request on 32-bit input capture ch.2
		1	Interrupt request on 32-bit input capture ch.2
bit12	ICIR3	0	No interrupt request on 32-bit input capture ch.3
		1	Interrupt request on 32-bit input capture ch.3
bit11 to bit8	Undefined	"0" is read.	

11.3.6 Interrupt Request Batch-Read Register 3 Lower (IRPR3L)

Interrupt vector number 37 (decimal) is used for the following peripheral functions:

- UART/CSIO/I^{2}C ch.7 receive interrupt request
- 32-bit input capture ch.4 to ch.7

This register can be read to check the peripheral function from which an interrupt request has been generated.

Figure 11.3-6 shows the bit configuration of interrupt request batch-read register 3 lower (IRPR3L).

Figure 11.3-6 Bit configuration of interrupt request batch-read register 3 lower (IRPR3L)

bit	7	6	5	4	3	2	1	0
	ICIR4	ICIR5	ICIR6	ICIR7	RXIR7	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	ICIR4	0	No interrupt request on 32-bit input capture ch.4
		1	Interrupt request on 32-bit input capture ch.4
bit6	ICIR5	0	No interrupt request on 32-bit input capture ch.5
		1	Interrupt request on 32-bit input capture ch.5
bit5	ICIR6	0	No interrupt request on 32-bit input capture ch.6
		1	Interrupt request on 32-bit input capture ch.6
bit4	ICIR7	0	No interrupt request on 32-bit input capture ch.7
		1	Interrupt request on 32-bit input capture ch.7
bit3	RXIR7	0	No receive interrupt request on UART/CSIO/I^{2}C ch.7
		1	Receive interrupt request on UART/CSIO/I^{2}C ch.7
bit2 to bit0	Undefined	"0" is read.	

11.3.7 Interrupt Request Batch-Read Register 4 Upper (IRPR4H)

Interrupt vector number 45 (decimal) is used for 32-bit output compare channels ch.0 to ch.3. This register can be read to check on which channel an interrupt request has been generated.

Figure 11.3-7 shows the bit configuration of interrupt request batch-read register 4 upper (IRPR4H).

Figure 11.3-7 Bit configuration of Interrupt request batch-read register 4 upper (IRPR4H)

bit	15	14	13	12	11	10	9	8
	OCIR0	OCIR1	OCIR2	OCIR3	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	OCIR0	0	No interrupt request on 32-bit output compare ch.0
		1	Interrupt request on 32-bit output compare ch.0
bit14	OCIR1	0	No interrupt request on 32-bit output compare ch.1
		1	Interrupt request on 32-bit output compare ch.1
bit13	OCIR2	0	No interrupt request on 32-bit output compare ch.2
		1	Interrupt request on 32-bit output compare ch.2
bit12	OCIR3	0	No interrupt request on 32-bit output compare ch.3
		1	Interrupt request on 32-bit output compare ch.3
bit11 to bit8	Undefined	"0" is read.	

11.3.8 Interrupt Request Batch-Read Register 4 Lower (IRPR4L)

Interrupt vector number 38 (decimal) is used for the following peripheral functions:

- UART/CSIO/I^2C ch.7 transmit/transmit bus idle
- I^2C ch.7 status interrupt request
- 32-bit output compare ch.4 to ch.7

This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-8 shows the bit configuration of interrupt request batch-read register 4 lower (IRPR4L).

Figure 11.3-8 Bit configuration of interrupt request batch-read register 4 lower (IRPR4L)

bit	7	6	5	4	3	2	1	0
	OCIR4	OCIR5	OCIR6	OCIR7	TXIR7	ISIR7	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	OCIR4	0	No interrupt request on 32-bit output compare ch.4
		1	Interrupt request on 32-bit output compare ch.4
bit6	OCIR5	0	No interrupt request on 32-bit output compare ch.5
		1	Interrupt request on 32-bit output compare ch.5
bit5	OCIR6	0	No interrupt request on 32-bit output compare ch.6
		1	Interrupt request on 32-bit output compare ch.6
bit4	OCIR7	0	No interrupt request on 32-bit output compare ch.7
		1	Interrupt request on 32-bit output compare ch.7
bit3	TXIR7	0	No UART/CSIO/I^2C ch.7 transmit/transmit bus idle
		1	UART/CSIO/I^2C ch.7 transmit/transmit bus idle
bit2	ISIR7	0	No I^2C ch.7 status interrupt request
		1	I^2C ch.7 status interrupt request
bit1, bit0	Undefined	"0" is read.	

11.3.9 Interrupt Request Batch-Read Register 5 Upper (IRPR5H)

Interrupt vector number 50 (decimal) is used for base timer channels ch.4 and ch.5. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-9 shows the bit configuration of interrupt request batch-read register 5 upper (IRPR5H).

Figure 11.3-9 Bit configuration of Interrupt request batch-read register 5 upper (IRPR5H)

bit	15	14	13	12	11	10	9	8
	BT0IR4	BT1IR4	BT0IR5	BT1IR5	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	BT0IR4	0	No interrupt request 0 generated on base timer ch.4
		1	Interrupt request 0 generated on base timer ch.4
bit14	BT1IR4	0	No interrupt request 1 generated on base timer ch.4
		1	Interrupt request 1 generated on base timer ch.4
bit13	BT0IR5	0	No interrupt request 0 generated on base timer ch.5
		1	Interrupt request 0 generated on base timer ch.5
bit12	BT1IR5	0	No interrupt request 1 generated on base timer ch.5
		1	Interrupt request 1 generated on base timer ch.5
bit11 to bit8	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.10 Interrupt Request Batch-Read Register 5 Lower (IRPR5L)

Interrupt vector number 51 (decimal) is used for base timer channels ch.6 and ch.7. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-10 shows the bit configuration of interrupt request batch-read register 5 lower (IRPR5L).

Figure 11.3-10 Bit configuration of interrupt request batch-read register 5 lower (IRPR5L)

bit	7	6	5	4	3	2	1	0
	BT0IR6	BT1IR6	BT0IR7	BT1IR7	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	BT0IR6	0	No interrupt request 0 generated on base timer ch.6
		1	Interrupt request 0 generated on base timer ch.6
bit6	BT1IR6	0	No interrupt request 1 generated on base timer ch.6
		1	Interrupt request 1 generated on base timer ch.6
bit5	BT0IR7	0	No interrupt request 0 generated on base timer ch.7
		1	Interrupt request 0 generated on base timer ch.7
bit4	BT1IR7	0	No interrupt request 1 generated on base timer ch.7
		1	Interrupt request 1 generated on base timer ch.7
bit3 to bit0	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.11 Interrupt Request Batch-Read Register 6 Upper (IRPR6H)

Interrupt vector number 52 (decimal) is used for base timer channels ch.8 and ch.9. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-11 shows the bit configuration of interrupt request batch-read register 6 upper (IRPR6H).

Figure 11.3-11 Bit configuration of Interrupt request batch-read register 6 upper (IRPR6H)

bit	15	14	13	12	11	10	9	8
	BT0IR8	BT1IR8	BT0IR9	BT1IR9	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	BT0IR8	0	No interrupt request 0 generated on base timer ch.8
		1	Interrupt request 0 generated on base timer ch.8
bit14	BT1IR8	0	No interrupt request 1 generated on base timer ch.8
		1	Interrupt request 1 generated on base timer ch.8
bit13	BT0IR9	0	No interrupt request 0 generated on base timer ch.9
		1	Interrupt request 0 generated on base timer ch.9
bit12	BT1IR9	0	No interrupt request 1 generated on base timer ch.9
		1	Interrupt request 1 generated on base timer ch.9
bit11 to bit8	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.12 Interrupt Request Batch-Read Register 6 Lower (IRPR6L)

Interrupt vector number 53 (decimal) is used for base timer channels ch.10 and ch.11. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-12 shows the bit configuration of interrupt request batch-read register 6 lower (IRPR6L).

Figure 11.3-12 Bit configuration of interrupt request batch-read register 6 lower (IRPR6L)

bit	7	6	5	4	3	2	1	0
	BT0IR10	BT1IR10	BT0IR11	BT1IR11	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit7	BT0IR10	0	No interrupt request 0 generated on base timer ch.10
		1	Interrupt request 0 generated on base timer ch.10
bit6	BT1IR10	0	No interrupt request 1 generated on base timer ch.10
		1	Interrupt request 1 generated on base timer ch.10
bit5	BT0IR11	0	No interrupt request 0 generated on base timer ch.11
		1	Interrupt request 0 generated on base timer ch.11
bit4	BT1IR11	0	No interrupt request 1 generated on base timer ch.11
		1	Interrupt request 1 generated on base timer ch.11
bit3 to bit0	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.13 Interrupt Request Batch-Read Register 7 Upper (IRPR7H)

Interrupt vector number 56 (decimal) is used for base timer channels ch.14 and ch.15, and slave interface. This register can be read to check on which channels interrupt requests have been generated and the types of interrupt requests.

Figure 11.3-13 shows the bit configuration of interrupt request batch-read register 7 upper (IRPR7H).

Figure 11.3-13 Bit configuration of interrupt request batch-read register 7 upper (IRPR7H)

bit	15	14	13	12	11	10	9	8
	BT0IR14	BT1IR14	BT0IR15	BT1IR15	Undefined	SLIR	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When an interrupt request is generated, the bit corresponding to the generated interrupt request is set to "1".

Bit number	Bit	Value	Explanation
bit15	BT0IR14	0	No interrupt request 0 generated on base timer ch.14
		1	Interrupt request 0 generated on base timer ch.14
bit14	BT1IR14	0	No interrupt request 1 generated on base timer ch.14
		1	Interrupt request 1 generated on base timer ch.14
bit13	BT0IR15	0	No interrupt request 0 generated on base timer ch.15
		1	Interrupt request 0 generated on base timer ch.15
bit12	BT1IR15	0	No interrupt request 1 generated on base timer ch.15
		1	Interrupt request 1 generated on base timer ch.15
bit11	Undefined	"0" is read.	
bit10	SLIR	0	No interrupt request on the slave interface.
		1	Interrupt request on the slave interface.
bit9, bit8	Undefined	"0" is read.	

Interrupt requests 0 and 1 vary depending on the mode of the base timer operation.

Modes of base timer operation	Interrupt request 0	Interrupt request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

11.3.14 Interrupt Request Batch-Read Register 7 Lower (IRPR7L)

Interrupt vector number 61 (decimal) is used for DMA controller (DMAC) channels ch.4 to ch.7. This register can be read to check on which channel an interrupt request has been generated.

Figure 11.3-14 shows the bit configuration of interrupt request batch-read register 7 lower (IRPR7L).

Figure 11.3-14 Bit configuration of Interrupt request batch-read register 7 lower (IRPR7L)

bit	7	6	5	4	3	2	1	0
	DMAC4	DMAC5	DMAC6	DMAC7	Undefined	Undefined	Undefined	Undefined
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

When one of the following interrupt requests is generated on DMA controller (DMAC) ch.4 to ch.7, the bit corresponding to the generated interrupt request is set to "1".

- Normal end interrupt request
- Abnormal end interrupt request
- Transfer stop interrupt request

Bit number	Bit	Value	Explanation
bit7	DMAC4	0	No interrupt request on DMAC ch.4
		1	Interrupt request on DMAC ch.4
bit6	DMAC5	0	No interrupt request on DMAC ch.5
		1	Interrupt request on DMAC ch.5
bit5	DMAC6	0	No interrupt request on DMAC ch.6
		1	Interrupt request on DMAC ch.6
bit4	DMAC7	0	No interrupt request on DMAC ch.7
		1	Interrupt request on DMAC ch.7
bit3 to bit0	Undefined	"0" is read.	

11.4 Notes on Use

Note the following points about using the interrupt request batch-read function.

■ Notes on operations

- Writing to the interrupt request batch-read register (IRPR0 to IRPR7) is disabled. To cancel an interrupt request, clear the interrupt request flag bit of the corresponding function register.

CHAPTER 12 Delay Interrupt

This chapter explains the functions and operations of the delay interrupt function.

12.1 Overview

12.2 Configuration

12.3 Registers

12.4 An Explanation of Operations and Setting Procedure Examples

12.5 Notes on Use

12.1 Overview

The delay interrupt function generates task switching interrupts used by a real-time OS.

■ Overview

The delay interrupt function generates task switching interrupt requests used by a real-time OS such as REALOS. Software can use delay interrupts to generate interrupt requests to the CPU or cancel them.

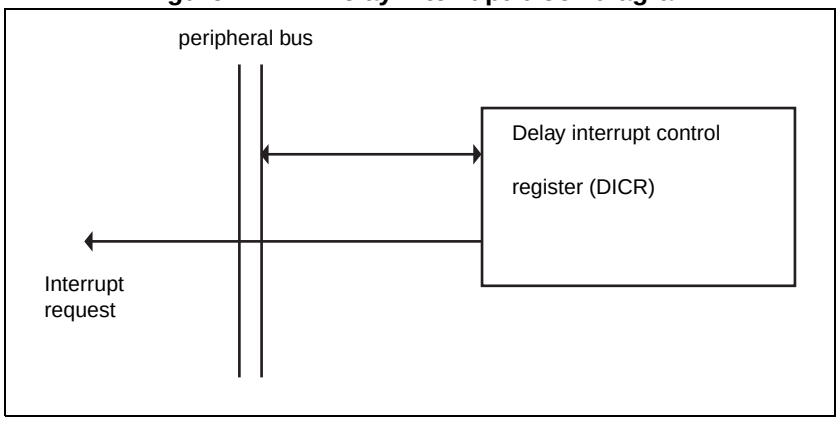
12.2 Configuration

This section explains the configuration of delay interrupts.

■ Delay interrupt block diagram

Figure 12.2-1 shows a delay interrupt block diagram.

Figure 12.2-1 Delay interrupt block diagram



- Delayed interrupt control register (DICR)
  This register controls delay interrupts.

■ Clocks

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

12.3 Registers

This section explains the configuration and functions of the register used for delay interrupts.

■ Delay interrupt register

Table 12.3-1 shows the delay interrupt register.

Table 12.3-1 Delay interrupt register

Abbreviated Register Name	Register Name	Reference
DICR	Delayed interrupt control register	12.3.1

12.3.1 Delayed Interrupt Control Register (DICR)

This register controls delay interrupts.

Figure 12.3-1 shows the bit configuration of the delayed interrupt control register (DICR).

Figure 12.3-1 Bit configuration of delayed interrupt control register (DICR)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	DLYI
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	1	1	1	1	1	1	0

R/W: Read/Write

[bit7 to bit1]: Undefined bits

In case of writing	Ignored
In case of reading	"1" is read.

[bit0]: DLYI (delay interrupt control bit)

This bit is used to enable generation of delay interrupt requests or cancel the delay interrupt requests.

Written Value	Explanation
0	Cancels delay interrupt source or generates no delay interrupt request
1	Generation of delay interrupt requests.

<Note>

This bit is used in the same way as other interrupt request flags. Clear this bit in the interrupt processing routine and switch tasks accordingly.

12.4 An Explanation of Operations and Setting Procedure Examples

This section explains delay interrupt operations and the setting procedure for delay interrupts.

12.4.1 Explanation of Delay Interrupt Operations

Software can use delay interrupts to generate interrupt requests to the CPU or cancel them.
Table 12.4-1 lists the conditions for generating delay interrupts.

Table 12.4-1 Interrupt request generation conditions

Interrupt request	Delay interrupt request
Interrupt request generation	Write "1" to the DLYI bit of the delayed interrupt control register (DICR).
Interrupt request enabled	None (interrupts always enabled)
Clearing an interrupt request	Write "0" to the DLYI bit of the delayed interrupt control register (DICR).

<Notes>

- Delay interrupts cannot be used for DMA transfer requests.
- For information on interrupt vector numbers, see "APPENDIX C Interrupt Vectors".
- Use an interrupt control register (ICR47) to specify the interrupt level corresponding to the interrupt vector number.For information on the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

12.5 Notes on Use

Note the following points about using delay interrupts.

■ Notes on the program

- The delay interrupt control bit can be used in the same way as other interrupt request flags. Clear this bit in the interrupt routine and switch tasks accordingly.
- Delay interrupts cannot be used for DMA transfer requests.

CHAPTER 13 External Bus Interface

This chapter explains the functions and operations of the external bus interface.

13.1 Overview
13.2 Configuration
13.3 Pins
13.4 Registers
13.5 Protocols
13.6 Timing Settings
13.7 Access Cycle Extension Using the RDY Pin
13.8 Number of Access Cycles
13.9 Address Information and Address Alignment
13.10 Data Alignment
13.11 External Bus DMA Transfer
13.12 CS Area Setting Procedure

13.1 Overview

The external bus interface connects this device with external machines (memory, I/O, and other devices) to input and output data.

■ Overview

The external bus interface has the following features:

- Address information of up to 24 bits long (32-MB address space, maximum with address shift) can be output.
- One of the following bus types can be selected:
  - Address data split bus
    Access destination address information is output only to the address bus.
    Asynchronous memory can be connected.
  - Address data multiplex bus
    Access destination address information is output to both the address bus and data bus.
- The following settings can be made individually for each of the 4 chip select areas (CS areas):
  - CS area size: A value ranging from 64 KB to 32 MB can be set.
  - CS area location: Any location in the external bus area can be set.
- The chip select settings corresponding to each CS area can be output.
- The following settings can be made in each CS area:
  - Whether operations are valid or invalid
  - Data bus width (8 bits/16 bits)
  - Whether the write operation is enabled or disabled (Disabled: Used as a read-only area)
  - Byte ordering (big endian/little endian)
    * Only big endian can be specified for the CS0 area.
  - Address type (normal output/address shift output)
  - Bus type (address data split bus/address data multiplex bus)
- The settings described below for periods (number of cycles) can be made in each CS area.
  Settings common to read access and write access
  - Chip select delay cycle
    Period from output of an address to enabling of the chip select
  - Address strobe output cycle
    Address strobe validity period
  - Access cycle
    Read/Write access cycle extension using the ready input pin
  - Address output cycle
    Period in which the data output pins output address information (only for a multiplex bus)

  Settings for read access
  - Read access automatic wait
    Read strobe validity period
  - Read access setup cycle
    Period from output of the chip select to enabling of the read strobe

- Read access hold cycle
  Period from disabling of the read strobe to disabling of the chip select
- Read access idle cycle
  Idle period after read access

Settings for write access

- Write access automatic wait
  Write strobe validity period
- Write access setup cycle
  Period from enabling of the chip select to enabling of the write strobe
- Write access hold cycle
  Period from disabling of the write strobe to disabling of the chip select
- Write recovery cycle
  Idle period after write access

• DMA transfer is supported with external bus pins.

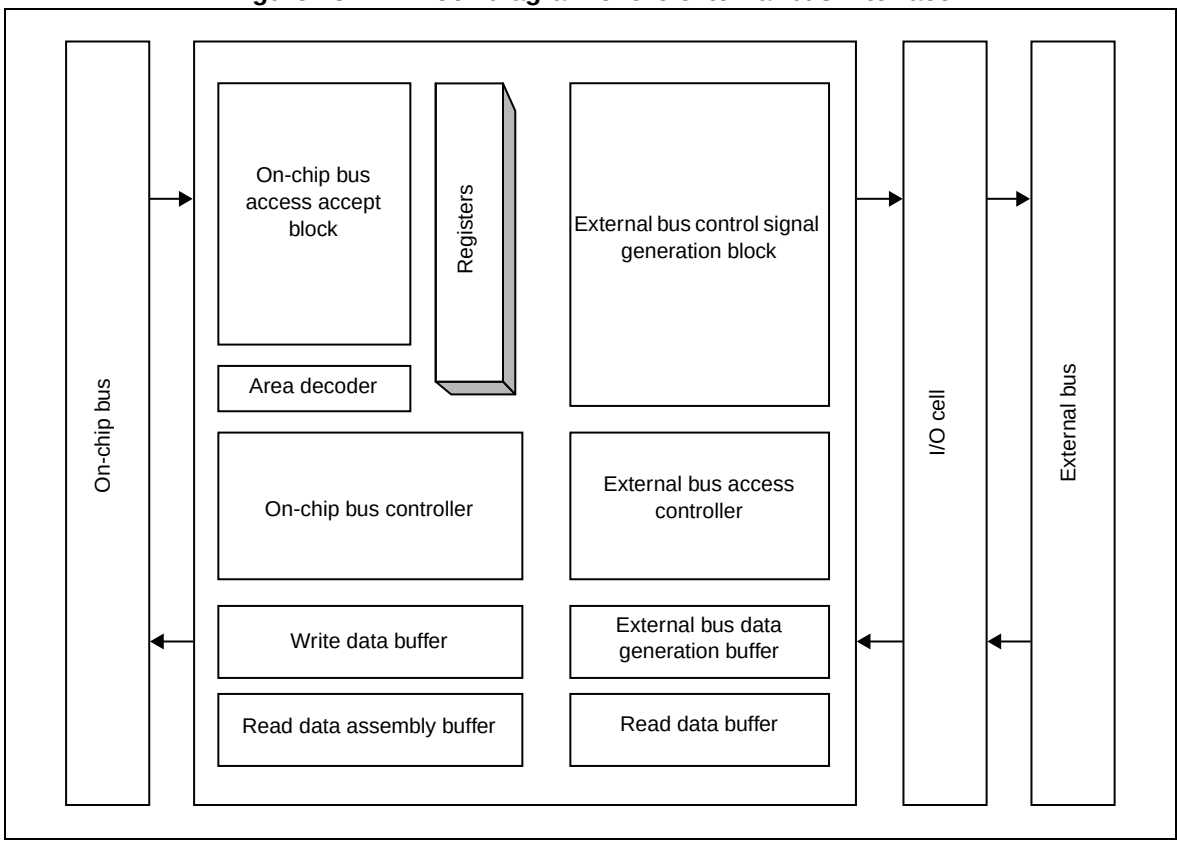
13.2 Configuration

This section explains the configuration of the external bus interface.

■ Block diagram of the external bus interface

Figure 13.2-1 is a block diagram of the external bus interface.

Figure 13.2-1 Block diagram of the external bus interface



- On-chip bus access accept block
  Accepts access requests to the external bus interface from the on-chip bus.
- Area decoder block
  Determines which CS area is accessed.
- On-chip bus controller
  Controls the on-chip bus.
- Write data buffer
  Stores data for output to an external machine in write access.
- Read data assembly buffer
  Assembles data that was first written by an external machine, split into parts and then input to the external bus interface in read access.
- External bus control signal generation block
  Generates address strobe, chip select, read strobe, write strobe, and other signals.

- External bus access controller

  Controls the output periods and output timing of address strobe, chip select, read strobe, write strobe, and other signals.
- External bus data generation buffer

  Splits data that will be output to an external machine into parts according to the bus width in write access.
- Read data buffer

  Stores data written by an external device in read access.

■ Clocks

Table 13.2-1 lists the clock used with the external bus interface.

Table 13.2-1 Clock used with the external bus interface

Clock Name	Description	Remarks
Operation clock	External bus clock (TCLK)	Internal operation clock

A clock with the same frequency as that of the external bus clock (TCLK) can be output to the SYSCLK pin as a bus clock.

13.3 Pins

This section explains the pins of the external bus interface.

■ Overview

- A23 to A00 pins

  Address output pins of the external bus interface. These pins are used as an address bus, and they output access destination address information.

  These pins are multiplexed pins. For details of using the A23 to A00 pins of the external bus interface, see "2.4 Setting Method for Pins".

- D15 to D00 pins

  Data I/O pins of the external bus interface. These pins are used as a data bus.

  These pins are multiplexed pins. For details of using the D15 to D00 pins of the external bus interface, see "2.4 Setting Method for Pins".

- $\overline{CS0}$ to $\overline{CS3}$ pins

  External bus interface chip select output pins

  An external machine processes a request from the external bus interface during output of an "L" level signal by one of these pins.

  These pins are multiplexed pins. For details of using the $\overline{CS0}$ to $\overline{CS3}$ pins of the external bus interface, see "2.4 Setting Method for Pins".

- $\overline{AS}$ pin

  External bus interface address strobe output pin

  When outputting an "L" level signal, this pin operates as the address strobe, which indicates the start of bus access.

  This pin is a multiplexed pin. For details of using the $\overline{AS}$ pin of the external bus interface , see "2.4 Setting Method for Pins".

- $\overline{RD}$ pin

  External bus interface read strobe output pin

  An external machine transmits data through the D15 to D00 pins during output of an "L" level signal by this pin.

  This pin is a multiplexed pin. For details of using the $\overline{RD}$ pin of the external bus interface , see "2.4 Setting Method for Pins".

- $\overline{WR0}$ and $\overline{WR1}$ pins

  External bus interface write strobe output pins. These pins perform write operations by the byte.

  The write operation to an external machine can be performed during output of an "L" level signal by one of these pins.

  These pins are multiplexed pins. For details of using the $\overline{WR0}$ and $\overline{WR1}$ pins of the external bus interface, see "2.4 Setting Method for Pins".

- RDY pin

  External bus interface ready input pin. Inputting an "L" level signal through this pin extends the access cycle.

  This pin is a multiplexed pin. For details of using the RDY pin of the external bus interface, see "2.4 Setting Method for Pins".

- SYSCLK pin

  External bus interface bus clock output pin

  This pin is a multiplexed pin. For details of using the SYSCLK pin of the external bus interface, see "2.4 Setting Method for Pins".

The external bus interface uses also the following pins, which are DMA controller (DMAC) pins:

- DREQ0 to DREQ3 pins

  Input pins for DMA transfer requests
- DACK0 to DACK3 pins

  Output pins for acceptance signals of DMA transfer requests
- DEOP0 to DEOP3 pins

  Output pins for DMA transfer end signals

13.4 Registers

This section explains the configuration and functions of registers of the external bus interface.

■ External bus interface registers

Table 13.4-1 lists the registers of the external bus interface.

Table 13.4-1 External bus interface registers

CS Area	Abbreviated Register Name	Register Name	Reference
0	ASR0	CS0 area setting register	13.4.1
	ACR0	CS0 area configuration register	13.4.2
	AWR0	CS0 area wait register	13.4.3
	DMAR0	DMA transfer register 0	13.4.4
1	ASR1	CS1 area setting register	13.4.1
	ACR1	CS1 area configuration register	13.4.2
	AWR1	CS1 area wait register	13.4.3
	DMAR1	DMA transfer register 1	13.4.4
2	ASR2	CS2 area setting register	13.4.1
	ACR2	CS2 area configuration register	13.4.2
	AWR2	CS2 area wait register	13.4.3
	DMAR2	DMA transfer register 2	13.4.4
3	ASR3	CS3 area setting register	13.4.1
	ACR3	CS3 area configuration register	13.4.2
	AWR3	CS3 area wait register	13.4.3
	DMAR3	DMA transfer register 3	13.4.4

13.4.1 Area Setting Registers (ASR0 to ASR3)

These registers specify the CS areas (CS0 to CS3). Each of these registers is provided for one corresponding CS area.

Figure 13.4-1 shows the bit configuration of the area setting registers (ASR0 to ASR3).

Figure 13.4-1 Bit configuration of the area setting registers (ASR0 to ASR3)

bit	31							16
	SADR31 to SADR16							
Attribute	R/W							
Initial value (ASR0)	0							
Initial value (ASR1 to ASR3)	X							

bit	15							8
	Reserved							
Attribute	R/W							
Initial value (ASR0)	0							
Initial value (ASR1 to ASR3)	0							

bit	7	6	5	4	3	2	1	0
	ASZ3	ASZ2	ASZ1	ASZ0	Reserved	WREN	LEDN	CSEN
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (ASR0)	1	1	1	1	0	0	0	1
Initial value (ASR1 to ASR3)	X	X	X	X	0	X	X	0

R/W: Read/Write

X: Undefined

<Notes>

- Be sure that the CS areas are not set to overlap one other.
- For details of setting these registers, see "13.12 CS Area Setting Procedure".
- The initial value of the CS0 area setting register (ASR0) differs from those of the CS1 to CS3 area setting registers (ASR1 to ASR3).
- Be sure to write data to these registers in units of words.

[bit31 to bit16]: SADR31 to SADR16 (Start address specification bits)

These bits specify the start address of a CS area.

The upper 16 bits of a 32-bit address must be specified for these bits.

The range that begins from the address specified by these bits is allocated as a CS area with the size specified by the ASZ3 to ASZ0 bits.

<Note>

The boundary of a CS area depends on the size specified by the ASZ3 to ASZ0 bits. Therefore, the bits that are actually compared with an address vary depending on the size specified by the ASZ3 to ASZ0 bits. To determine which bits are actually compared, see the ASZ3 to ASZ0 bits.

[bit15 to bit8]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit7 to bit4]: ASZ3 to ASZ0 (Area size bits)

These bits set the size of a CS area.

Among the SADR31 to SADR16 bits, the bits that are actually compared with an address is determined according to the size specified by these bits.

ASZ3	ASZ2	ASZ1	ASZ0	CS Area Size	Bits Compared with Address
0	0	0	0	64 KB	SADR31 to SADR16 bits
0	0	0	1	128 KB	SADR31 to SADR17 bits
0	0	1	0	256 KB	SADR31 to SADR18 bits
0	0	1	1	512 KB	SADR31 to SADR19 bits
0	1	0	0	1 MB	SADR31 to SADR20 bits
0	1	0	1	2 MB	SADR31 to SADR21 bits
0	1	1	0	4 MB	SADR31 to SADR22 bits
0	1	1	1	8 MB	SADR31 to SADR23 bits
1	0	0	0	16 MB	SADR31 to SADR24 bits

<Note>

Do not make any settings except those described in the table.

[bit3]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2]: WREN (Write enable bit)

This bit enables/disables the write operation to a CS area from an external machine.

Written Value	Explanation
0	Disables write.
1	Enables write.

<Note>

While the write operation is disabled, any write operation from the internal bus to a CS area is ignored, and no external access is possible.

[bit1]: LEDN (Little endian setting bit)

This bit selects either big endian or little endian for the byte ordering of a CS area.

This bit in the CS0 area setting register (ASR0) is treated as an undefined bit since the CS0 area supports only big endian.

- In the CS0 area setting register (ASR0)

In case of writing	Ignored
In case of reading	"0" is read.

- In the CS1 to CS3 area setting registers (ASR1 to ASR3)

Written Value	Explanation
0	Big endian
1	Little endian

[bit0]: CSEN (CS Area enable bit)

This bit enables/disables a CS area.

When enabled by this bit, the CS area starts operating according to the settings of this register and the following registers:

- Area setting register (ASR0 to ASR3)
- Area configuration register (ACR0 to ACR3)
- Area wait register (AWR0 to AWR3)

Written Value	Explanation
0	Disables the CS area.
1	Enables the CS area.

13.4.2 Area Configuration Registers (ACR0 to ACR3)

These registers specify the bus for the CS areas (CS0 to CS3). Each of these registers is provided for one corresponding CS area.

Figure 13.4-2 shows the bit configuration of the area configuration registers (ACR0 to ACR3).

Figure 13.4-2 Bit configuration of the area configuration registers (ACR0 to ACR3)

bit	31							8
	Reserved							
Attribute	R/W							
Initial value (ACR0)	0							
Initial value (ACR1 to ACR3)	0							

bit	7	6	5	4	3	2	1	0
	DBW1	DBW0	Reserved	Reserved	ADTY	BSTY	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (ACR0)	0	0	0	0	0	0	0	0
Initial value (ACR1 to ACR3)	X	X	0	0	X	X	0	X

R/W: Read/Write

X: Undefined

<Notes>

- For details of setting these registers, see "13.12 CS Area Setting Procedure".
- The initial value of the CS0 area configuration register (ACR0) differs from those of the CS1 to CS3 area configuration registers (ACR1 to ACR3).
- Be sure to write data to these registers in units of words.

[bit31 to bit8]: Reserved bits

In case of writing	Always write "0" to this bit (these bits).
In case of reading	"0" is read.

[bit7, bit6]: DBW1, DBW0 (Data bus width bits)

These bits set the data bus width.

DBW1	DBW0	Data Bus Width
0	0	8 bits
0	1	16 bits
1	0	Setting prohibited
1	1	Setting prohibited

The data bus width specified by these bits determines which data bus and write strobe output pins are used.

Table 13.4-2 lists data bus widths and the corresponding pins used.

Table 13.4-2 Data bus widths and the corresponding pins used

Data Bus Width	Data Bus	Write Strobe Output Pin
8 bits	D15 to D08	$\overline{WR0}$
16 bits	D15 to D00	$\overline{WR0}$, $\overline{WR1}$

[bit5, bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit3]: ADTY (Address type bit)

This bit specifies one of the following methods for output of access destination address information:

- Normal output: Access destination address information is output without any modification.
- Address shift output: Access destination address information is output after a bit shift is applied.

For details of such address information, see "13.9 Address Information and Address Alignment".

Written Value	Explanation
0	Normal output
1	Address shift output

[bit2]: BSTY (Bus type bit)

When the address output is address shift output mode (ADTY=1), this bit specifies a bus type. Either output of 24-bit address information from only the address bus (A23 to A00 pins) or output of such information to both the address bus (A23 to A00 pins) and data bus (D15 to D00 pins) can be set.

- Address data split bus: Output of 24-bit address information from only the address bus (A23 to A00 pins).
- Address data multiplex bus: Output of access destination address information to both the address bus (A23 to A00 pins) and data bus (D15 to D00 pins).

Written Value	Explanation
0	Address data split bus
1	Address data multiplex bus

Table 13.4-3 lists the pins that output address information during the address output cycle to set the address data multiplex bus.

Table 13.4-3 Address information that is output and the pins used

BSTY	Data Bus Width	Address Information Output	Pins That Output Address Information
1	8 bits	bit7 to bit0 of address information	D15 to D08 pins
	16 bits	bit15 to bit0 of address information	D15 to D00 pins

[bit1]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: Reserved bit

(ACR0)

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

(ACR1 to ACR3)

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	An initial values is undefined. "0" is read after reading "0".

13.4.3 Area Wait Registers (AWR0 to AWR3)

These registers set CS area wait and signal output periods. Each of these registers is provided for one corresponding CS area.

Figure 13.4-3 shows the bit configuration of the area wait registers (AWR0 to AWR3).

Figure 13.4-3 Bit configuration of the area wait registers (AWR0 to AWR3)

bit	31	30	29	28	27	26	25	24
	Reserved	Reserved	Reserved	Reserved	RWT3	RWT2	RWT1	RWT0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (AWR0)	0	0	0	0	1	1	1	1
Initial value (ACR1 to ACR3)	0	0	0	0	X	X	X	X
bit	23	22	21	20	19	18	17	16
	WWT3	WWT2	WWT1	WWT0	RIDL1	RIDL0	WRCV1	WRCV0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (AWR0)	0	0	0	0	0	0	0	0
Initial value (ACR1 to ACR3)	X	X	X	X	X	X	X	X
bit	15	14	13	12	11	10	9	8
	CSRD1	CSRD0	RDCS1	RDCS0	CSWR1	CSWR0	WRCS1	WRCS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (AWR0)	1	1	1	1	0	0	0	0
Initial value (ACR1 to ACR3)	X	X	X	X	X	X	X	X
bit	7	6	5	4	3	2	1	0
	ADCY1	ADCY0	ACS1	ACS0	ASCY	Reserved	RDYE	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value (AWR0)	0	0	0	0	0	0	0	0
Initial value (ACR1 to ACR3)	X	X	X	X	X	0	X	0

R/W: Read/Write

X: Undefined

<Notes>

- For details of setting these registers, see "13.12 CS Area Setting Procedure".
- The initial value of the CS0 area wait register (AWR0) differs from those of the CS1 to CS3 area wait registers (AWR1 to AWR3).
- Be sure to write data to these registers in units of words.

[bit31 to bit28]: Reserved bits

In case of writing	Always write "0" to this bit (these bits).
In case of reading	"0" is read.

[bit27 to bit24]: RWT3 to RWT0 (Read access automatic wait bits)

These bits specify the read strobe output period (read access automatic wait).
The read strobe is output for a period of at least 1T (T: Bus clock period).

RWT3	RWT2	RWT1	RWT0	Explanation
0	0	0	0	0T
0	0	0	1	1T
0	0	1	0	2T
0	0	1	1	3T
0	1	0	0	4T
0	1	0	1	5T
0	1	1	0	6T
0	1	1	1	7T
1	0	0	0	8T
1	0	0	1	9T
1	0	1	0	10T
1	0	1	1	11T
1	1	0	0	12T
1	1	0	1	13T
1	1	1	0	14T
1	1	1	1	15T

T: Bus clock period

[bit23 to bit20]: WWT3 to WWT0 (Write access automatic wait bits)

These bits specify the write strobe output period (write access automatic wait).

The write strobe is output for a period of at least 1T (T: Bus clock period).

WWT3	WWT2	WWT1	WWT0	Explanation
0	0	0	0	0T
0	0	0	1	1T
0	0	1	0	2T
0	0	1	1	3T
0	1	0	0	4T
0	1	0	1	5T
0	1	1	0	6T
0	1	1	1	7T
1	0	0	0	8T
1	0	0	1	9T
1	0	1	0	10T
1	0	1	1	11T
1	1	0	0	12T
1	1	0	1	13T
1	1	1	0	14T
1	1	1	1	15T

T: Bus clock period

[bit19, bit18]: RIDL1, RIDL0 (Read access idle cycle bits)

These bits specify the idle cycles (read access idle cycles) that are inserted after read access.

If the access immediately after read access is any of the following, as many idle cycles as specified by these bits are inserted after the read access:

- Write access
- Access to another CS area
- Access to a CS area for which the address data multiplex bus is set as the bus type

RIDL1	RIDL0	Explanation
0	0	0T
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Notes>

- Since all chip select signals are disabled ("H" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins) and the D15 to D00 pins become Hi-Z during a read access idle cycle, the next access does not begin until the read access idle cycle ends.
- No read access idle cycle is inserted during continuous read access of one CS area for which the address data split bus is set as the bus type by the BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) in the corresponding area configuration register (ACR0 to ACR3).

[bit17, bit16]: WRCV1, WRCV0 (Write recovery cycle bits)

These bits specify the idle cycles (write recovery cycles) that are inserted after write access.

As many idle cycles as specified by these bits are inserted after an external machine reads data from this device.

WRCV1	WRCV0	Explanation
0	0	0T
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Note>

Since all chip select signals are disabled ("H" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins) and the write strobe is also disabled ("H" level output from the $\overline{WR0}$ and $\overline{WR1}$ pins) during write recovery cycles, the next access does not begin.

[bit15, bit14]: CSRD1, CSRD0 (Read access setup cycle bits)

These bits set the period to enable the read strobe after the chip select is enabled (read access setup cycle).

CSRD1	CSRD0	Explanation
0	0	0T (same time)
0	1	After 1T
1	0	After 2T
1	1	After 3T

T: Bus clock period

<Note>

If the address data multiplex bus is set as the bus type by the BSTY bit (ADTY=1 and BSTY=1) in the corresponding area configuration register (ACR0 to ACR3), settings must satisfy the following condition in conformity with the protocol:

- **ACS + CSRD ≥ 1**

  ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits

[bit13, bit12]: RDCS1, RDCS0 (Read access hold cycle bits)

These bits set the period to disable the chip select after the read strobe is disabled (read access hold cycle).

RDCS1	RDCS0	Explanation
0	0	0T (same time)
0	1	After 1T
1	0	After 2T
1	1	After 3T

T: Bus clock period

[bit11, bit10]: CSWR1, CSWR0 (Write access setup cycle bits)

These bits set the period to enable the write strobe after the chip select is enabled (write access setup cycle).

CSWR1	CSWR0	Explanation
0	0	0T (same time)
0	1	After 1T
1	0	After 2T
1	1	After 3T

T: Bus clock period

<Note>

If the address data multiplex bus is set as the bus type by the BSTY bit (ADTY=1 and BSTY=1) in the corresponding area configuration register (ACR0 to ACR3), settings must satisfy the following condition in conformity with the protocol:

- **ACS + CSWR ≥ 1**

  ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

[bit9, bit8]: WRCS1, WRCS0 (Write access hold cycle bits)

These bits set the period to disable the chip select after the write strobe is disabled (write access hold cycle).

WRCS1	WRCS0	Explanation
0	0	0T (same time)
0	1	After 1T
1	0	After 2T
1	1	After 3T

T: Bus clock period

[bit7, bit6]: ADCY1, ADCY0 (Address output extension cycle count bits)

These bits specify the period of address information output from the D15 to D00 pins, when the address data multiplex bus is set as the bus type (ADTY=1 and BSTY=1) (number of address output extension cycles).

The period in which address information is output from the D15 to D00 pins (address output cycle) is at least 1T (T: Bus clock period).

ADCY1	ADCY0	Explanation
0	0	0T
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Notes>

- If the address data split bus is set as the bus type by the BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) in the corresponding area configuration register (ACR0 to ACR3), the settings of these bits are ignored.
- To set a value other than "00" in these bits, settings must satisfy all of the following conditions in conformity with the protocol:
  - **ADCY + 1 ≤ ACS + CSRD**
    ADCY: ADCY1, ADCY0 bits ACS: ACS1, ACS0 bits
    CSRD: CSRD1, CSRD0 bits
  - **ADCY + 1 ≤ ACS + CSWR**
    ADCY: ADCY1, ADCY0 bits ACS: ACS1, ACS0 bits
    CSWR: CSWR1, CSWR0 bits
- If the period set by the ASCY1 and ASCY0 bits is greater than that set by these bits, the setting of the ASCY1 and ASCY0 bits takes priority.

[bit5, bit4]: ACS1, ACS0 (Chip select delay cycle count bits)

These bits set the period to enable the chip select ("L" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins) after output of the address strobe (number of chip select delay cycles).

ACS1	ACS0	Explanation
0	0	0T
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Note>

If the address data multiplex bus is set as the bus type by the BSTY bit (ADTY=1 and BSTY=1) in the corresponding area configuration register (ACR0 to ACR3), settings must satisfy all of the following conditions in conformity with the protocol:

- **ACS + CSRD ≥ 1**
  ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
- **ACS + CSWR ≥ 1**
  ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

[bit3]: ASCY (Address strobe output extension cycle count bit)

This bit specifies the address strobe output period (number of address strobe output extension cycles).

The address strobe is output for a period of at least 1T (T: Bus clock period).

Written Value	Explanation
0	0T
1	1T

T: Bus clock period

<Note>

To set "1" in this bit, settings must satisfy all of the following conditions in conformity with the protocol:

- If the address data split bus is set as the bus type by the BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) in the corresponding area configuration register (ACR0 to ACR3)
  - **ACS + CSRD + RWT + RDCS ≥ 1**
    ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
    RWT: RWT3 to RWT0 bits RDCS: RDCS1, RDCS0 bits
  - **ACS + CSWR + WWT + WRCS ≥ 1**
    ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits
    WWT: WWT3 to WWT0 bits WRCS: WRCS1, WRCS0 bits
- If the address data multiplex bus is set as the bus type by the BSTY bit (ADTY=1 and BSTY=1) in the corresponding area configuration register (ACR0 to ACR3)
  - **ACS + CSRD ≥ 2**
    ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
  - **ACS + CSWR ≥ 2**
    ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

[bit2]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit1]: RDYE (RDY enable bit)

This bit specifies whether to enable the automatic wait period extension function that uses the RDY pin.

Written Value	Explanation
0	Disabled
1	Enabled

<Note>

To enable this function, use the RWT3 to RWT0 bits and WWT3 to WWT0 bits to specify "2" or a higher value for the read access/write access automatic wait periods.

For details, see "13.7 Access Cycle Extension Using the RDY Pin".

[bit0]: Reserved bit

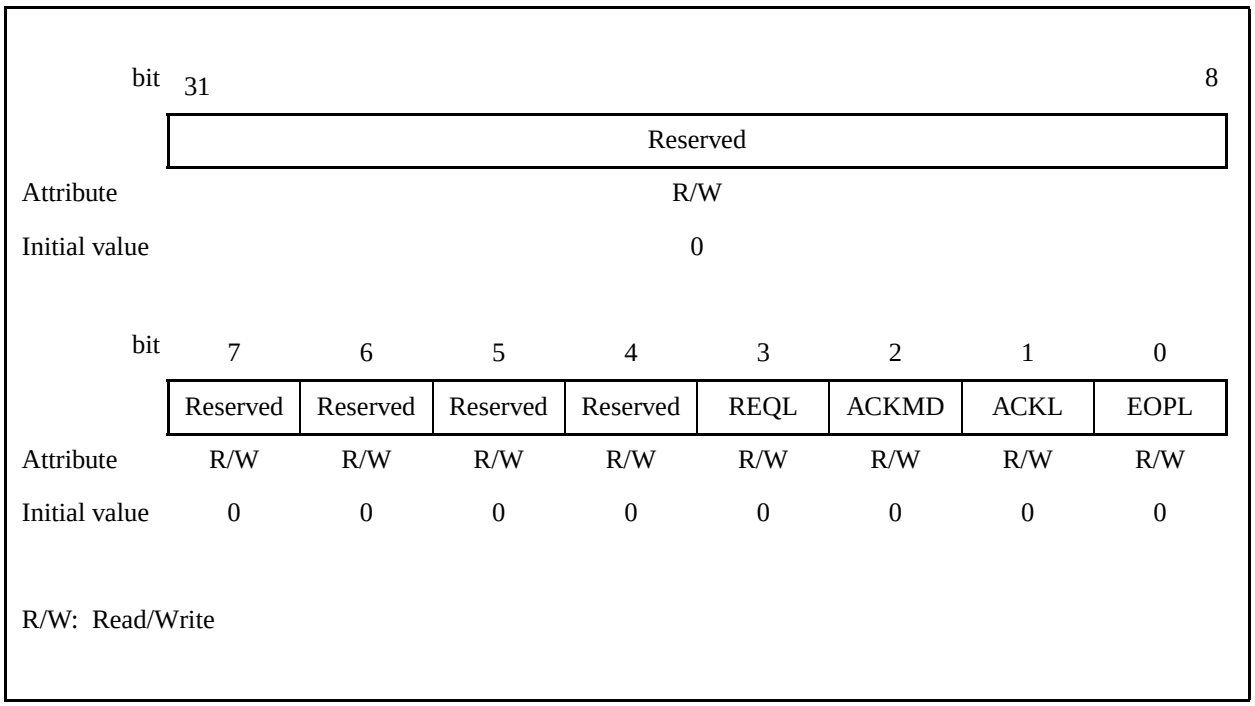
In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

13.4.4 DMA Transfer Registers (DMAR0 to DMAR3)

These registers set the external pins for DMA transfer. Each of these registers is provided for one corresponding DMA controller (DMAC) channel.

Figure 13.4-4 shows the bit configuration of the DMA transfer registers (DMAR0 to DMAR3).

Figure 13.4-4 Bit configuration of the DMA transfer registers (DMAR0 to DMAR3)


bit 31 8
Reserved
Attribute R/W
Initial value 0
bit 7 6 5 4 3 2 1 0
Reserved Reserved Reserved Reserved REQL ACKMD ACKL EOPL
Attribute R/W R/W R/W R/W R/W R/W R/W R/W
Initial value 0 0 0 0 0 0 0 0
R/W: Read/Write

<Note>

These registers must be set before the DMA controller (DMAC) starts operating. Also, when the DMA controller (DMAC) channel corresponding to one of these registers is operating, the value of the register must not be changed.

[bit31 to bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit3]: REQL (Transfer request level bit)

This bit specifies the level/edge for detecting a transfer request signal.

A transfer request signal is input from any of the DREQ0 to DREQ3 pins and reported to the DMA controller (DMAC).

The detection target level/edge varies depending on the setting for the DMA controller (DMAC) transfer method.

Written Value	Demand Transfer Time	Block Transfer/Burst Transfer
0	"L" level	Falling edge
1	"H" level	Rising edge

For details of the DMA transfer method, see "CHAPTER 33 DMA Controller (DMAC)".

[bit2]: ACKMD (Transfer request acceptance output mode bit)

This bit specifies the time that the DMA controller (DMAC) outputs a transfer request acceptance signal or transfer end signal from the DEOP0 to DEOP3 pins.

Written Value	Explanation
0	Outputs a signal in parallel with the chip select.
1	Outputs a signal in parallel with the read strobe/write strobe.

[bit1]: ACKL (Transfer request acceptance level bit)

This bit specifies the effective level of transfer request acceptance signals.

Output of such a signal at the level specified by this bit from the DACK0 to DACK3 pins indicates that the DMA controller (DMAC) has accepted a transfer request.

Written Value	Explanation
0	"L" level
1	"H" level

[bit0]: EOPL (Last transfer output level bit)

This bit specifies the effective level of transfer end signals.

Output of such a signal at the level specified by this bit from the DEOP0 to DEOP3 pins indicates that the DMA controller (DMAC) has finished a DMA transfer.

Written Value	Explanation
0	"L" level
1	"H" level

13.5 Protocols

This section explains the protocols for external bus interface signals.

13.5.1 Address Data Split Bus Protocol

This section explains the address data split bus protocol.

In the explanation of the protocol, the address data split bus is set as the bus type by the ADTY/BSTY bit (ADTY=0 or ADTY=1 and BSTY=0) in the corresponding area configuration register (ACR0 to ACR3).

■ Read protocol

● Read operation example

Figure 13.5-1 shows an example of operations in read access.

Figure 13.5-1 Example of operations in read access

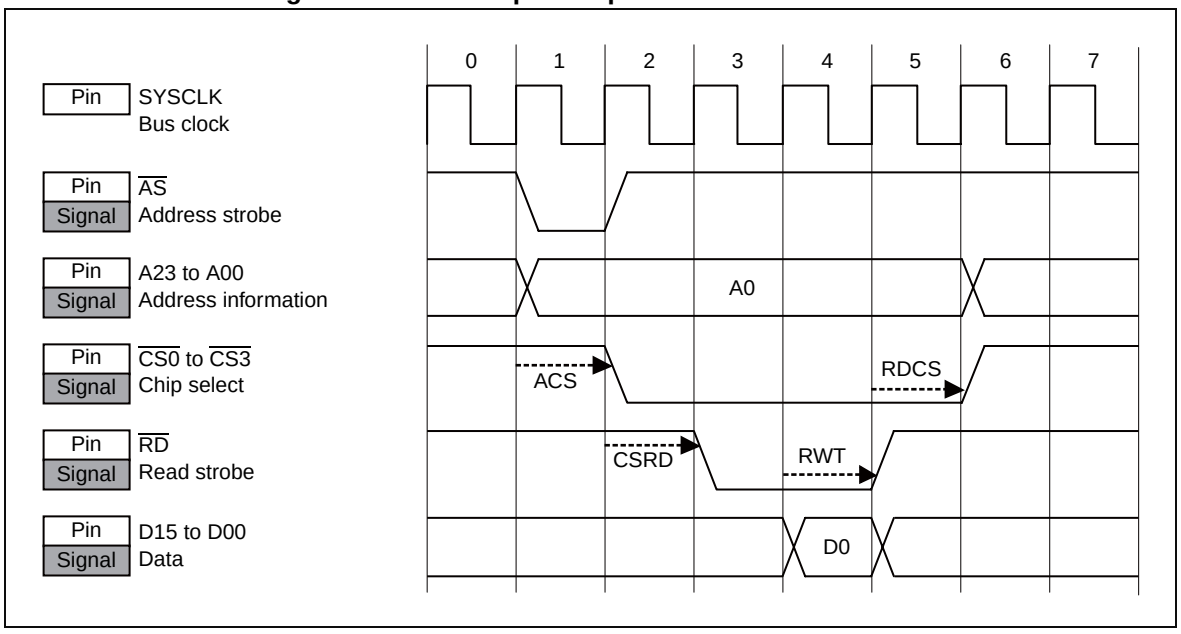


Table 13.5-1 lists the setting value of each bit in the area wait registers (AWR0 to AWR3).

Table 13.5-1 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Read access automatic wait	RWT3 to RWT0	0001
Read access idle cycle	RIDL1, RIDL0	00
Read access setup cycle	CSRD1, CSRD0	01
Read access hold cycle	RDCS1, RDCS0	01

● Shortest read operation

Figure 13.5-2 shows the shortest operation in read access.

Figure 13.5-2 Shortest read access

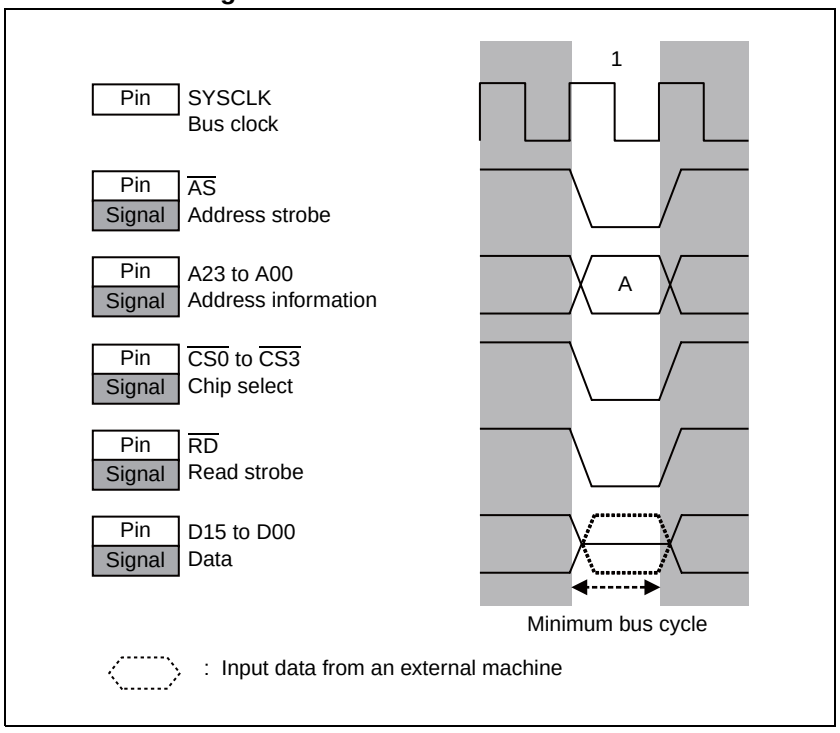


Table 13.5-2 lists the setting value of each bit in the area wait registers (AWR0 to AWR3) in the shortest read operation.

Table 13.5-2 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	00
Read access automatic wait	RWT3 to RWT0	0000
Read access idle cycle	RIDL1, RIDL0	00
Read access setup cycle	CSRD1, CSRD0	00
Read access hold cycle	RDCS1, RDCS0	00

● Explanation of signals

- SYSCLK pin

  This pin outputs the bus clock.
- $\overline{AS}$ pin

  This pin outputs the address strobe (valid at the "L" level). It indicates the start of access.
- A23 to A00 pins

  These pins output access destination address information.
- $\overline{CS0}$ to $\overline{CS3}$ pins

  These pins output the chip select (valid at the "L" level). This indicates that the access destination is the address in the corresponding CS area.
- $\overline{WR0}$ and $\overline{WR1}$ pins

  Output by these pins is at the "H" level (invalid).
- $\overline{RD}$ pin

  This pin outputs the read strobe (valid at the "L" level). It indicates read access.
- D15 to D00 pins

  These pins input data from an external machine.

● Access procedure

The read operation of the address data split bus follows the procedure below.

1. Enable the address strobe with the $\overline{AS}$ pin, and then output address information to the A23 to A00 pins.
2. Enable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.
3. Enable the read strobe with the $\overline{RD}$ pin.
4. Input read data from the D15 to D00 pins at the rising edge of the last bus clock within read strobe validity interval.
5. Disable the read strobe of the $\overline{RD}$ pin.
6. Disable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.

Output of address information to the A23 to A00 pins continues until the read operation is completed.

The output period and output timing of each signal can be changed through the settings of the area wait registers (AWR0 to AWR3). See "13.6 Timing Settings".

■ Write protocol

● Write operation example

Figure 13.5-3 shows an example of operations in write access.

Figure 13.5-3 Example of operations in write access

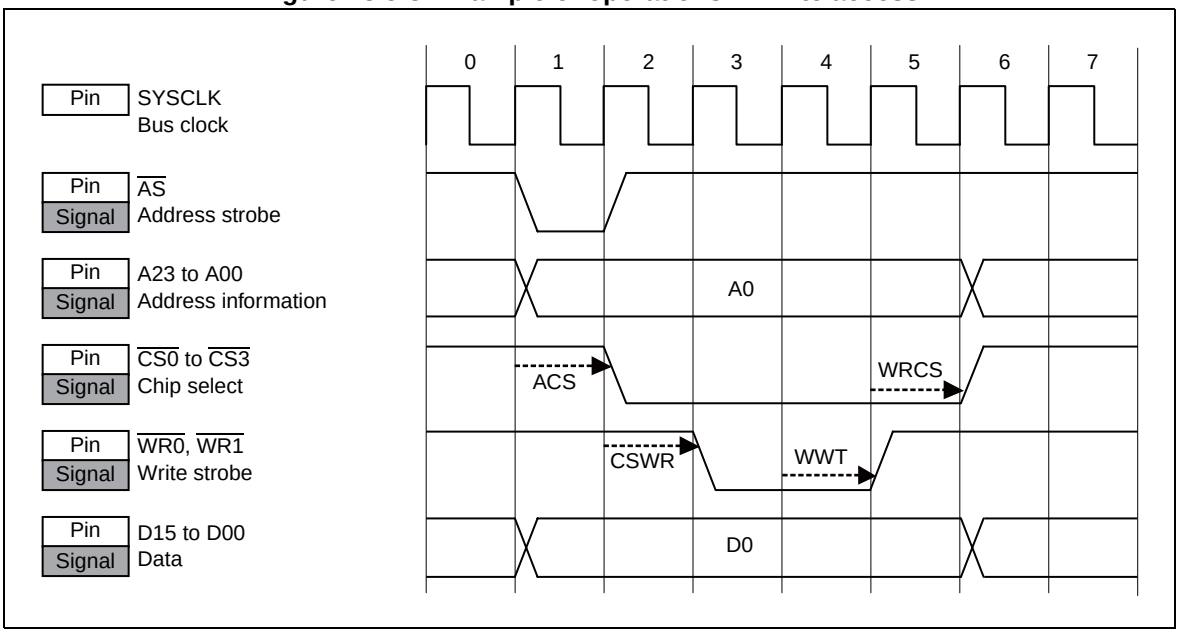


Table 13.5-3 lists the setting value of each bit in the area wait registers (AWR0 to AWR3).

Table 13.5-3 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Write access automatic wait	WWT3 to WWT0	0001
Write recovery cycle	WRCV1, WRCV0	00
Write access setup cycle	CSWR1, CSWR0	01
Write access hold cycle	WRCS1, WRCS0	01

● Shortest write operation

Figure 13.5-4 shows the shortest operation in write access.

Figure 13.5-4 Shortest write operation

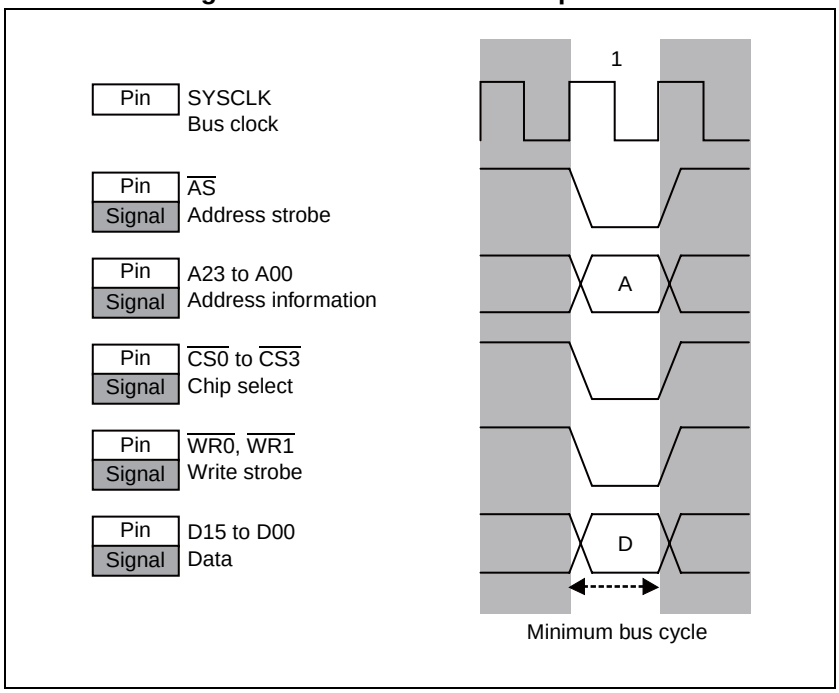


Table 13.5-4 lists the setting value of each bit in the area wait registers (AWR0 to AWR3) in the shortest write operation.

Table 13.5-4 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	00
Write access automatic wait	WWT3 to WWT0	0000
Write recovery cycle	WRCV1, WRCV0	00
Write access setup cycle	CSWR1, CSWR0	00
Write access hold cycle	WRCS1, WRCS0	00

● Explanation of signals

- SYSCLK pin

  This pin outputs the bus clock.
- $\overline{AS}$ pin

  This pin outputs the address strobe (valid at the "L" level). It indicates the start of access.
- A23 to A00 pins

  These pins output access destination address information.
- $\overline{CS0}$ to $\overline{CS3}$ pins

  These pins output the chip select (valid at the "L" level). This indicates that the access destination is the address in the corresponding CS area.
- $\overline{WR0}$ and $\overline{WR1}$ pins

  These pins output the write strobe (valid at the "L" level).
- $\overline{RD}$ pin

  Output by this pin is at the "H" level (invalid).
- D15 to D00 pins

  These pins output data to an external machine.

● Access procedure

The write operation of the address data split bus follows the procedure below.

1. Enable the address strobe with the $\overline{AS}$ pin, and then output address information to the A23 to A00 pins and write data to the D15 to D00 pins.
2. Enable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.
3. Enable the write strobe with the $\overline{WR0}$ and $\overline{WR1}$ pins.
4. Disable the write strobe with the $\overline{WR0}$ and $\overline{WR1}$ pins.
5. Disable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.

Output of address information to the A23 to A00 pins and write data to the D15 to D00 pins continues until the write operation is completed.

The output period and output timing of each signal can be changed through the settings of the area wait registers (AWR0 to AWR3). See "13.6 Timing Settings".

13.5.2 Address Data Multiplex Bus Protocol

This section explains the address data multiplex bus protocol.

In the explanation of the protocol, the address data multiplex bus is set as the bus type by the ADTY/ BSTY bit (ADTY=1 and BSTY=1) in the corresponding area configuration register (ACR0 to ACR3).

■ Read protocol

● Read operation example

Figure 13.5-5 shows an example of operations in read access.

Figure 13.5-5 Example of operations in read access

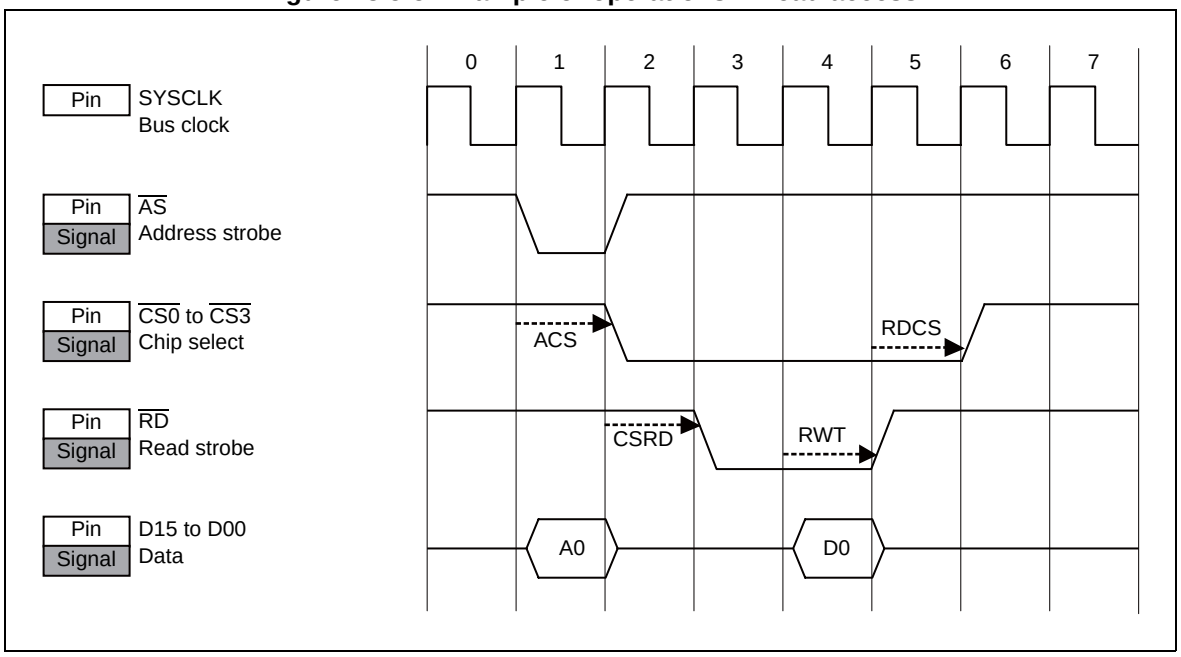


Table 13.5-5 lists the setting value of each bit in the area wait registers (AWR0 to AWR3).

Table 13.5-5 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Read access automatic wait	RWT3 to RWT0	0001
Read access idle cycle	RIDL1, RIDL0	00
Read access setup cycle	CSRD1, CSRD0	01
Read access hold cycle	RDCS1, RDCS0	01
Number of address output extension cycles	ADCY1, ADCY0	00

● Shortest read operation

Figure 13.5-6 shows one of the shortest operations in read access.

Figure 13.5-6 Shortest read access (ACS1 = 0, ACS0 = 1)

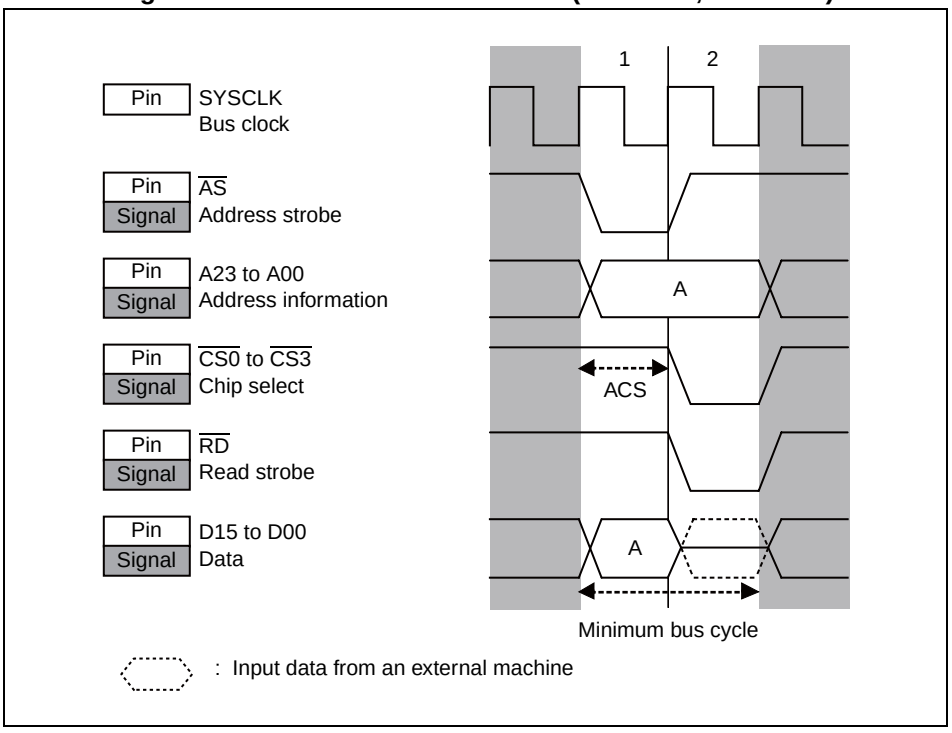


Table 13.5-6 lists the setting value of each bit in the area wait registers (AWR0 to AWR3) in the shortest read operation.

Table 13.5-6 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Read access automatic wait	RWT3 to RWT0	0000
Read access idle cycle	RIDL1, RIDL0	00
Read access setup cycle	CSRD1, CSRD0	00
Read access hold cycle	RDCS1, RDCS0	00
Number of address output extension cycles	ADCY1, ADCY0	00

The minimum bus cycle for the address data multiplex bus must be 2T (T: Bus clock period). Either the number of chip select delay cycles or number of read access setup cycles must be "1" or higher. Figure 13.5-6 and Table 13.5-6 shows that "01_B" is set as the number of chip select delay cycles (ACS1, ACS0).

● Explanation of signals

- SYSCLK pin

  This pin outputs the bus clock.
- $\overline{AS}$ pin

  This pin outputs the address strobe (valid at the "L" level). It indicates the start of access.
- A23 to A00 pins

  These pins output access destination address information.
- $\overline{CS0}$ to $\overline{CS3}$ pins

  These pins output the chip select (valid at the "L" level). This indicates that the access destination is the address in the corresponding CS area.
- $\overline{WR0}$ and $\overline{WR1}$ pins

  Output by this pin is at the "H" level (invalid).
- $\overline{RD}$ pin

  This pin outputs the read strobe (valid at the "L" level). It indicates read access.
- D15 to D00 pins

  These pins input data from an external machine after address information is output.

● Access procedure

The read operation of the address data multiplex bus follows the procedure below.

1. Enable the address strobe with the $\overline{AS}$ pin, and then output address information to the A23 to A00 pins and D15 to D00 pins.
2. Enable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.
3. Enable the read strobe with the $\overline{RD}$ pin.
4. Input read data from the D15 to D00 pins at the rising edge of the last bus clock within read strobe validity interval.
5. Disable the read strobe of the $\overline{RD}$ pin.
6. Disable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.

The address information on the D15 to D00 pins is the same as that on the A15 to A00 pins. The D15 to D00 pins are placed in the high impedance state after the address information is output. Output of address information to the A23 to A00 pins continues until the read operation is completed.

The output period and output timing of each signal can be changed through the settings of the area wait registers (AWR0 to AWR3). See "13.6 Timing Settings".

■ Write protocol

● Write operation example

Figure 13.5-7 shows an example of operations in write access.

Figure 13.5-7 Example of operations in write access

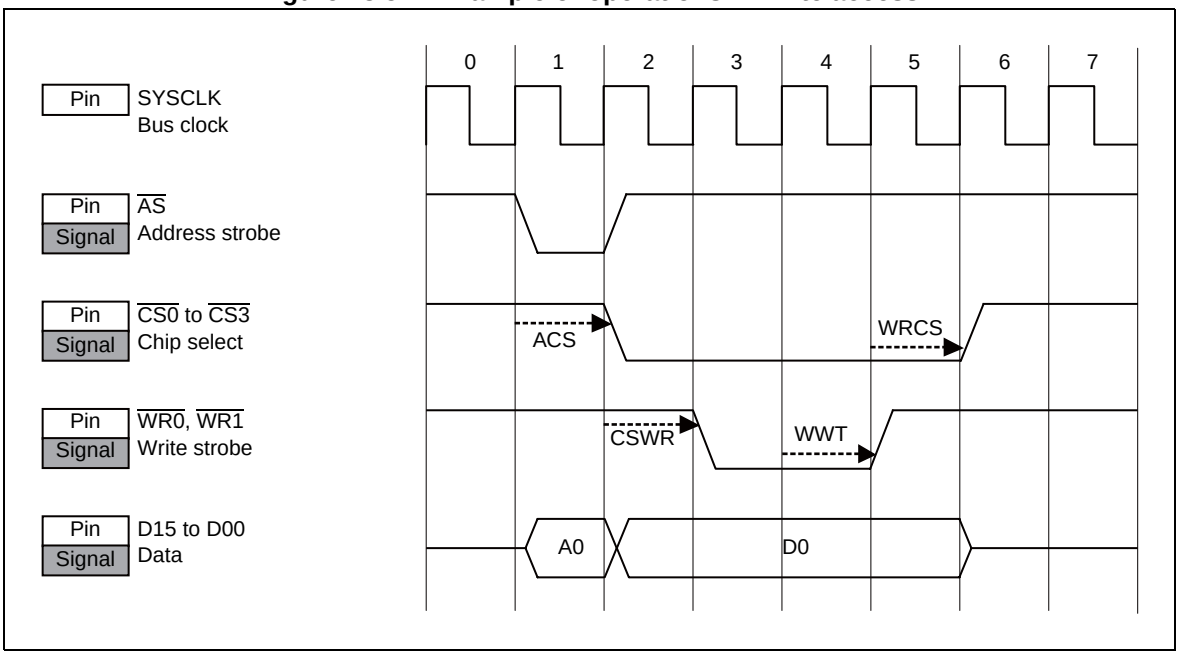


Table 13.5-7 lists the setting value of each bit in the area wait registers (AWR0 to AWR3).

Table 13.5-7 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Write access automatic wait	WWT3 to WWT0	0001
Write recovery cycle	WRCV1, WRCV0	00
Write access setup cycle	CSWR1, CSWR0	01
Write access hold cycle	WRCS1, WRCS0	01
Number of address output extension cycles	ADCY1, ADCY0	00

● Shortest write operation

Figure 13.5-8 shows one of the shortest operations in write access.

Figure 13.5-8 Shortest write access (ACS1 = 0, ACS0 = 1)

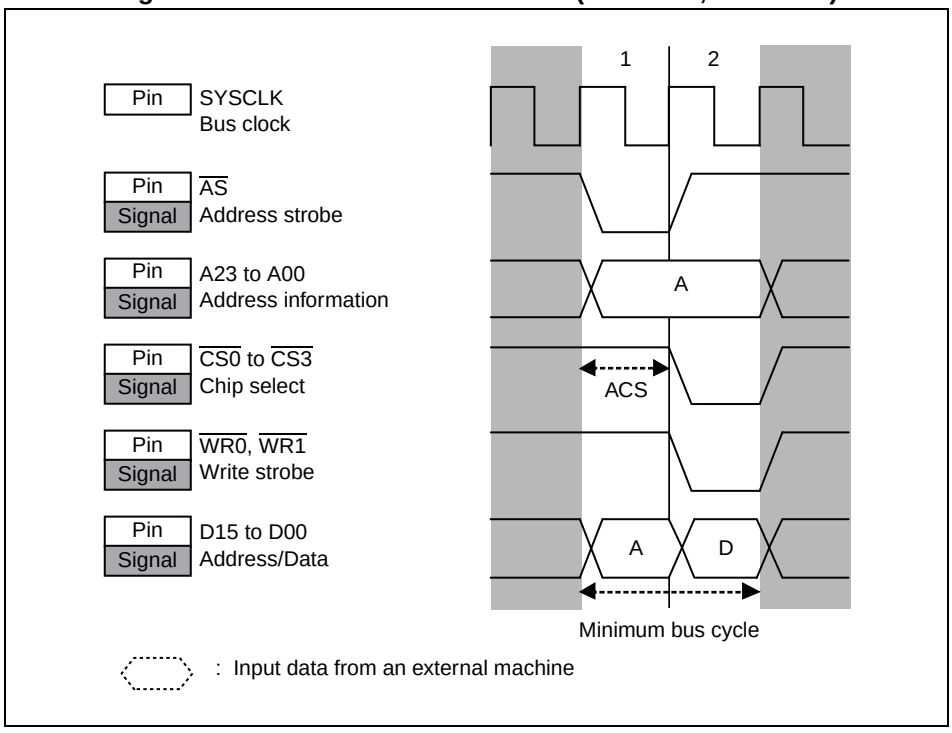


Table 13.5-8 lists the setting value of each bit in the area wait registers (AWR0 to AWR3) in the shortest write operation.

Table 13.5-8 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	0
Number of chip select delay cycles	ACS1, ACS0	01
Write access automatic wait	WWT3 to WWT0	0000
Write recovery cycle	WRCV1, WRCV0	00
Write access setup cycle	CSWR1, CSWR0	00
Write access hold cycle	WRCS1, WRCS0	00
Number of address output extension cycles	ADCY1, ADCY0	00

The minimum bus cycle for the address data multiplex bus must be 2T (T: Bus clock period). Either the number of chip select delay cycles or the number of write access setup cycles must be "1" or higher. Figure 13.5-8 and Table 13.5-8 shows that "01_B" is set as the number of chip select delay cycles (ACS1, ACS0).

● Explanation of signals

- SYSCLK pin

  This pin outputs the bus clock.
- $\overline{AS}$ pin

  This pin outputs the address strobe (valid at the "L" level). It indicates the start of access.
- A23 to A00 pins

  These pins output access destination address information.
- $\overline{CS0}$ to $\overline{CS3}$ pins

  These pins output the chip select (valid at the "L" level). This indicates the access destination is the address in the corresponding CS area.
- $\overline{WR0}$ and $\overline{WR1}$ pins

  These pins output the write strobe (valid at the "L" level). They indicate write access.
- $\overline{RD}$ pin

  Output by this pin is at the "H" level (invalid).
- D15 to D00 pins

  These pins input data from an external machine after address information is output.

● Access procedure

The write operation of the address data multiplex bus follows the procedure below.

1. Enable the address strobe with the $\overline{AS}$ pin, and then output address information to the A23 to A00 pins and D15 to D00 pins.
2. Output write data to the D15 to D00 pins.
3. Enable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.
4. Enable the write strobe with the $\overline{WR}$ pin.
5. Disable the write strobe with the $\overline{WR}$ pin.
6. Disable the chip select with the $\overline{CS0}$ to $\overline{CS3}$ pins.

The address information on the D15 to D00 pins is the same as that on the A15 to A00 pins. Output of address information to the A23 to A00 pins continues until the write operation is completed.

Output of write data to the D15 to D00 pins continues until the write operation is completed.

The output period and output timing of each signal can be changed through the settings of the area wait registers (AWR0 to AWR3). See "13.6 Timing Settings".

13.6 Timing Settings

This section explains the external bus interface timing settings. The output periods and output timing of signals can be set such that different types of external machines can be connected.

The following timing can be set with the bits of the area wait registers (AWR0 to AWR3):

- Read access automatic wait
- Read access idle cycle
- Read access setup cycle
- Write access setup cycle
- Chip select delay cycle
- Address strobe output extension cycle
- Write access automatic wait
- Write recovery cycle
- Read access hold cycle
- Write access hold cycle
- Address output extension cycle

<Note>

In order to help readers easily understand changes in timing through such settings, each period is set to the minimum value for the explanation of the basic protocol in this section.

Note that the settings for the basic protocol are different from the initial value settings.

13.6.1 Read Access Automatic Wait

The number of automatic wait cycles for read access is set. The read access automatic wait extends the read strobe validity period.

The period in which the read strobe remains valid is a minimum of 1T (T: Bus clock period), when no extension is applied. The length by which the period is extended can be specified as a period ranging from 0T to 15T (T: Bus clock period) with the RWT3 to RWT0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-1 lists the setting values of the RWT3 to RWT0 bits in the area wait registers (AWR0 to AWR3) and the corresponding output periods of the read strobe.

Table 13.6-1 Setting values of RWT3 to RWT0 and output periods of the read strobe

RWT3	RWT2	RWT1	RWT0	Extension Period	Read Strobe Output Period
0	0	0	1	0T	1T
0	0	0	1	1T	2T
0	0	1	0	2T	3T
0	0	1	1	3T	4T
0	1	0	0	4T	5T
0	1	0	1	5T	6T
0	1	1	0	6T	7T
0	1	1	1	7T	8T
1	0	0	0	8T	9T
1	0	0	1	9T	10T
1	0	1	0	10T	11T
1	0	1	1	11T	12T
1	1	0	0	12T	13T
1	1	0	1	13T	14T
1	1	1	0	14T	15T
1	1	1	1	15T	16T

T: Bus clock period

■ Address data split bus

Figure 13.6-1 shows an example in which the number of automatic wait cycles for read access is set to "1".

Figure 13.6-1 Example of read access automatic wait settings (address data split bus)

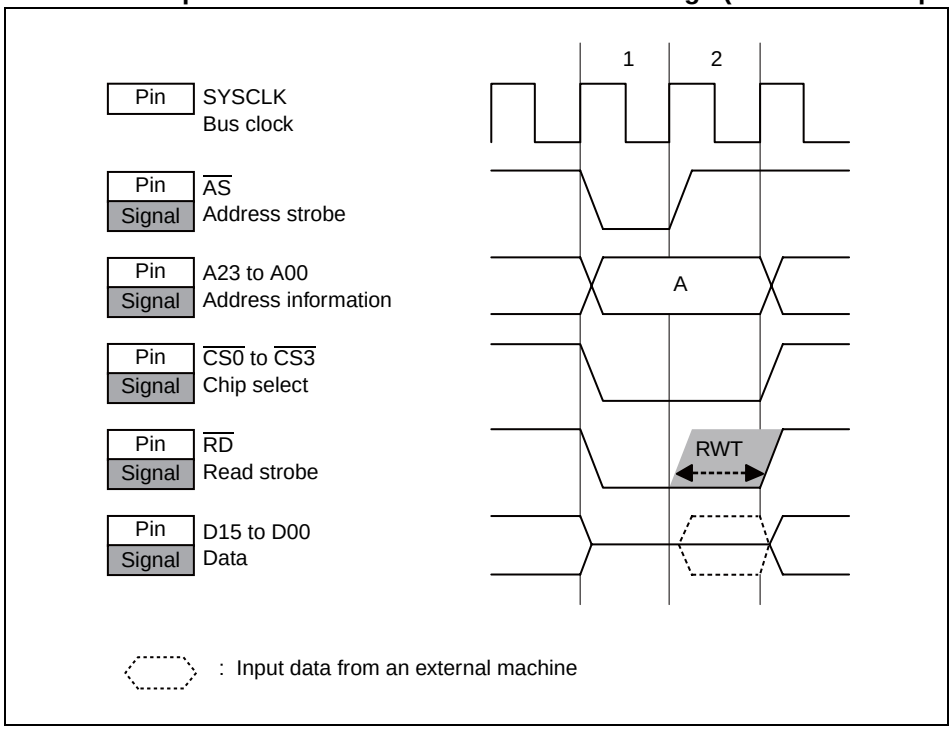


Table 13.6-2 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-1 (values other than "0" are set in the bits).

Table 13.6-2 Setting values of bits

Setting Item	Bit	Setting Value
Read access automatic wait	RWT3 to RWT0	0001

■ Address data multiplex bus

Figure 13.6-2 shows an example in which the read access automatic wait cycle is set to "1".

Figure 13.6-2 Example of read access automatic wait settings (address data multiplex bus)

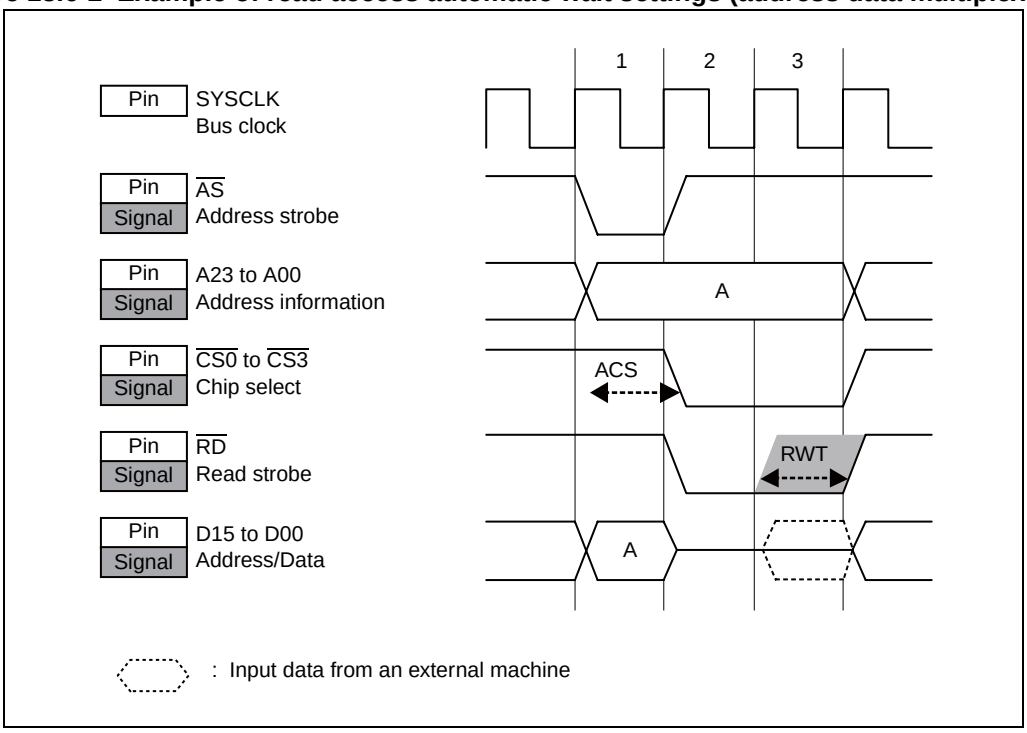


Table 13.6-3 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-2 (values other than "0" are set in the bits).

Table 13.6-3 Setting values of bits

Setting Item	Bit	Setting Value
Read access automatic wait	RWT3 to RWT0	0001
Number of chip select delay cycles	ACS1, ACS0	01

Figure 13.6-2 and Table 13.6-3 show that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.2 Write Access Automatic Wait

The number of automatic wait cycles for write access is set. The write access automatic wait cycle extends the write strobe validity period.

The period in which the write strobe remains valid is a minimum of 1T (T: bus clock cycle), when no extension is applied. The length by which the period is extended can be specified as a period ranging from 0T to 15T (T: Bus clock period) with the WWT3 to WWT0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-4 lists the setting values of the WWT3 to WWT0 bits in the area wait registers (AWR0 to AWR3) and the corresponding output periods of the write strobe.

Table 13.6-4 Setting values of WWT3 to WWT0 and output periods of the write strobe

WWT3	WWT2	WWT1	WWT0	Extension Period	Output Period (Total)
0	0	0	1	0T	1T
0	0	0	1	1T	2T
0	0	1	0	2T	3T
0	0	1	1	3T	4T
0	1	0	0	4T	5T
0	1	0	1	5T	6T
0	1	1	0	6T	7T
0	1	1	1	7T	8T
1	0	0	0	8T	9T
1	0	0	1	9T	10T
1	0	1	0	10T	11T
1	0	1	1	11T	12T
1	1	0	0	12T	13T
1	1	0	1	13T	14T
1	1	1	0	14T	15T
1	1	1	1	15T	16T

T: Bus clock period

■ Address data split bus

Figure 13.6-3 shows an example in which the write access automatic wait cycle is set to "1".

Figure 13.6-3 Example of write access automatic wait settings (address data split bus)

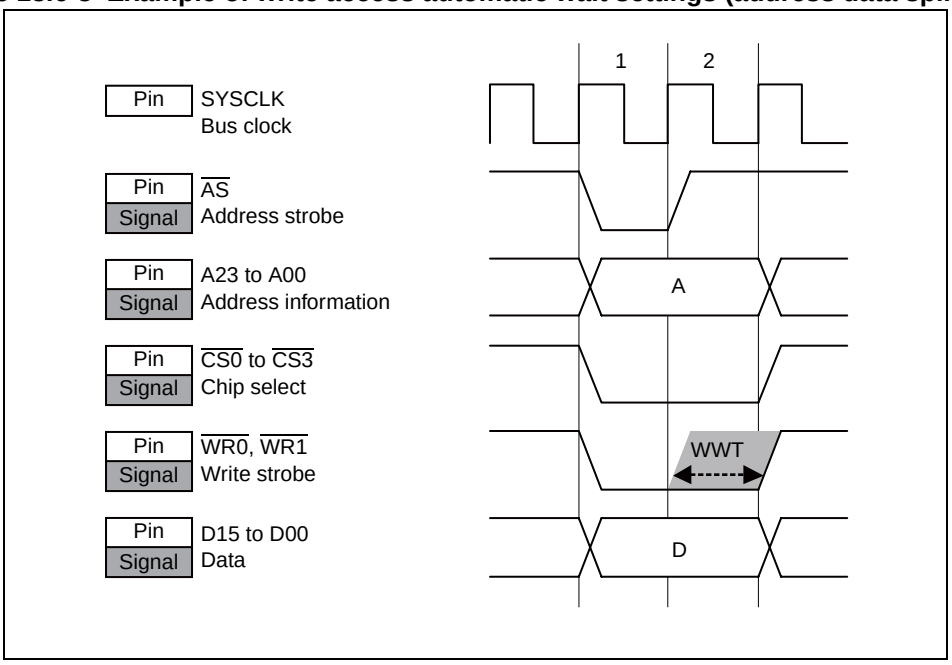


Table 13.6-5 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-3 (values other than "0" are set in the bits).

Table 13.6-5 Setting values of bits

Setting Item	Bit	Setting Value
Write access automatic wait	WWT3 to WWT0	0001

■ Address data multiplex bus

Figure 13.6-4 shows an example in which the write access automatic wait cycle is set to "1".

Figure 13.6-4 Example of write access automatic wait settings (address data multiplex bus)

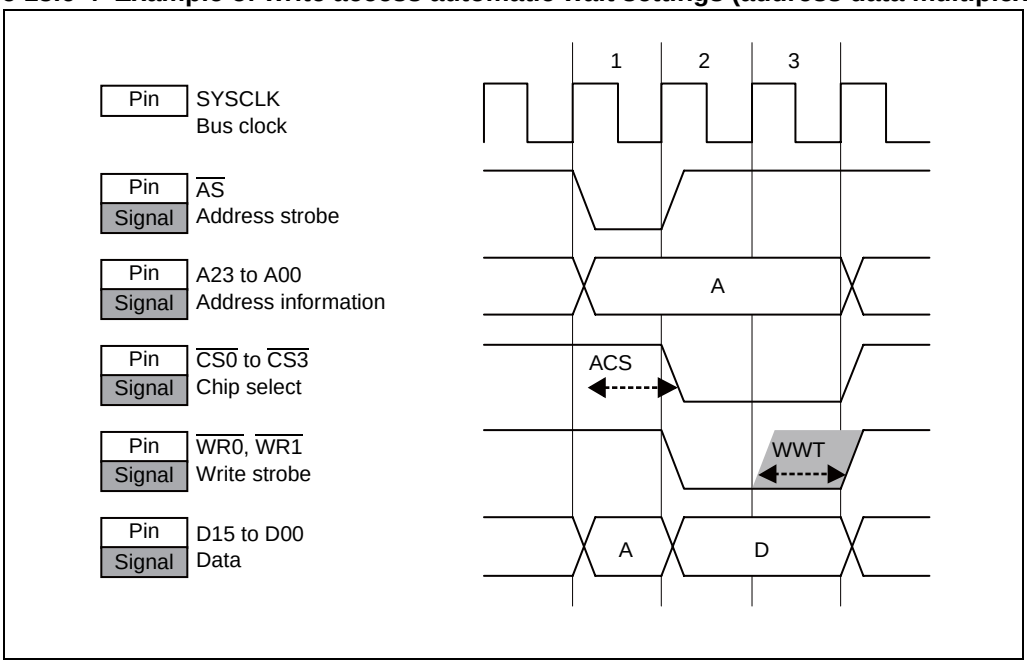


Table 13.6-6 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-4 (values other than "0" are set in the bits).

Table 13.6-6 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01
Write access automatic wait	WWT3 to WWT0	0001

Figure 13.6-4 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.3 Read Access Idle Cycle

The number of idle cycles for read access is set. If the read access idle cycle is set to "1" or higher, idle cycles are inserted after read access is completed.

All the chip select signals are invalid and the D15 to D00 pins are Hi-Z during the read access idle cycle. Inserting read access idle cycles prevents the read data received from an external machine that has a long output-off time from colliding with data associated with subsequent access on the bus.

If the next access after read access is any of the following, read access idle cycles are inserted after the read access is completed:

- Write access
- Access to another CS area
- Access to a CS area for which the address data multiplex bus is set as the bus type

The read access idle period can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the RIDL1 and RIDL0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-7 lists the setting values of the RIDL1 and RIDL0 bits in the area wait registers (AWR0 to AWR3) and the number of read access idle cycles.

Table 13.6-7 RIDL1 and RIDL0 bits and number of read access idle cycles

RIDL1	RIDL0	Number of Idle Cycles
0	0	0T (no read access idle)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Note>

No read access idle cycle is inserted during continuous read access of the same CS area of the address data split bus.

■ Address data split bus

Figure 13.6-5 shows an example in which the read access idle cycle is set to "1".

Figure 13.6-5 Example of read access idle cycle settings (address data split bus)

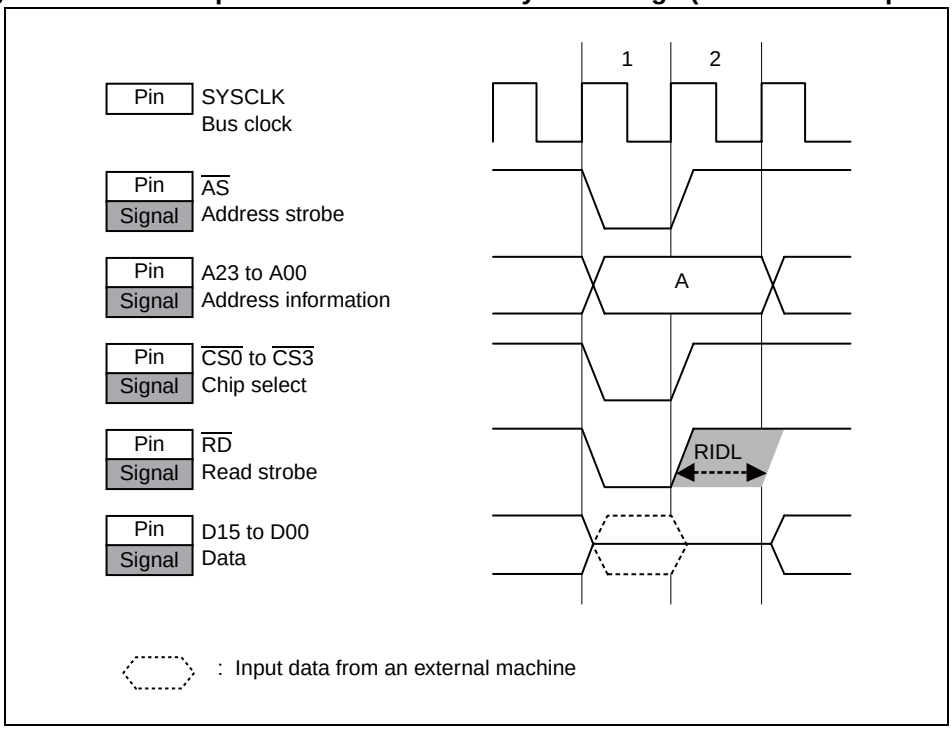


Table 13.6-8 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-5 (values other than "0" are set in the bits).

Table 13.6-8 Setting values of bits

Setting Item	Bit	Setting Value
Read access idle cycle	RIDL1, RIDL0	01

■ Address data multiplex bus

Figure 13.6-6 shows an example in which the number of idle cycles for read access is set to "1".

Figure 13.6-6 Example of read access idle cycle settings (address data multiplex bus)

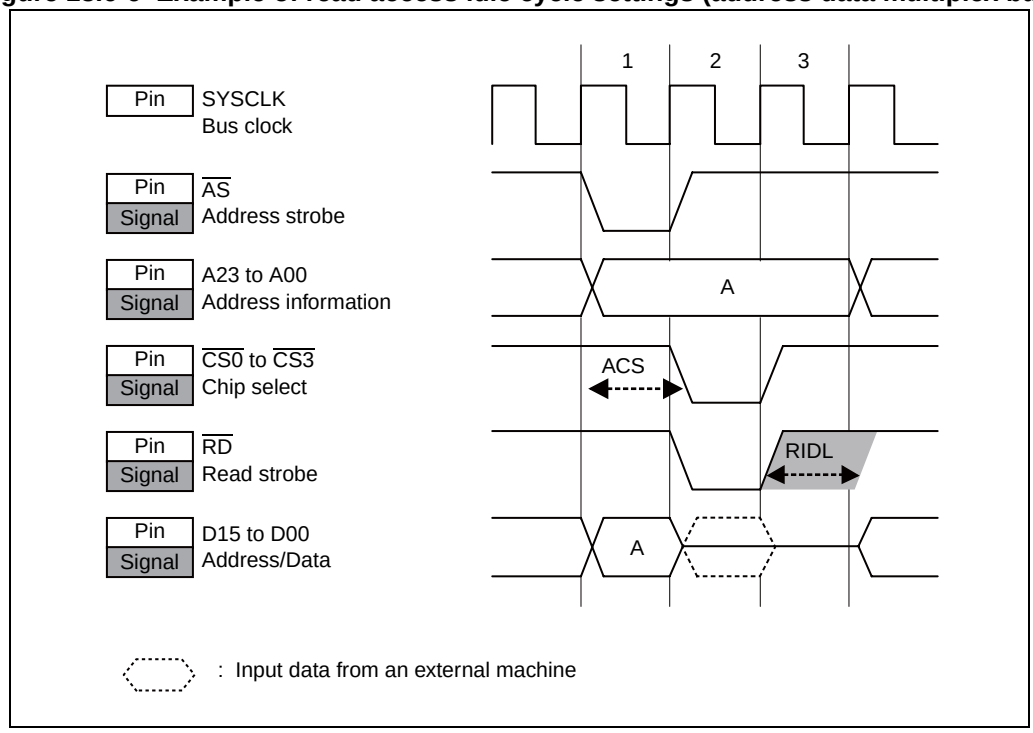


Table 13.6-9 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-6 (values other than "0" are set in the bits).

Table 13.6-9 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01
Read access idle cycle	RIDL1, RIDL0	01

Figure 13.6-6 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.4 Write Recovery Cycle

The number of write access recovery cycles is set. If the write recovery cycle is set to "1" or higher, recovery cycles are inserted after write access is completed.

All the chip select signals and the write strobe signal are invalid, and D15 to D00 pins become Hi-Z during write recovery cycles. If the next external machine to be accessed has restrictions between access operations, write recovery cycles are inserted after write access.

The write recovery cycle period can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the WRCV1 and WRCV0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-10 lists the setting values of the WRCV1 and WRCV0 bits in the area wait registers (AWR0 to AWR3) and the number of write recovery cycles.

Table 13.6-10 WRCV1, WRCV0 bits and number of write recovery cycles

WRCV1	WRCV0	Number of Write Recovery Cycles
0	0	0T (no write recovery)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Note>

If a value other than 0T (T: Bus clock period) is set as a write recovery cycle period, write recovery cycles are always inserted after write access.

■ Address data split bus

Figure 13.6-7 shows an example in which the write recovery cycle is set to 1T (T: Bus clock period).

Figure 13.6-7 Example of write recovery cycle settings (address data split bus)

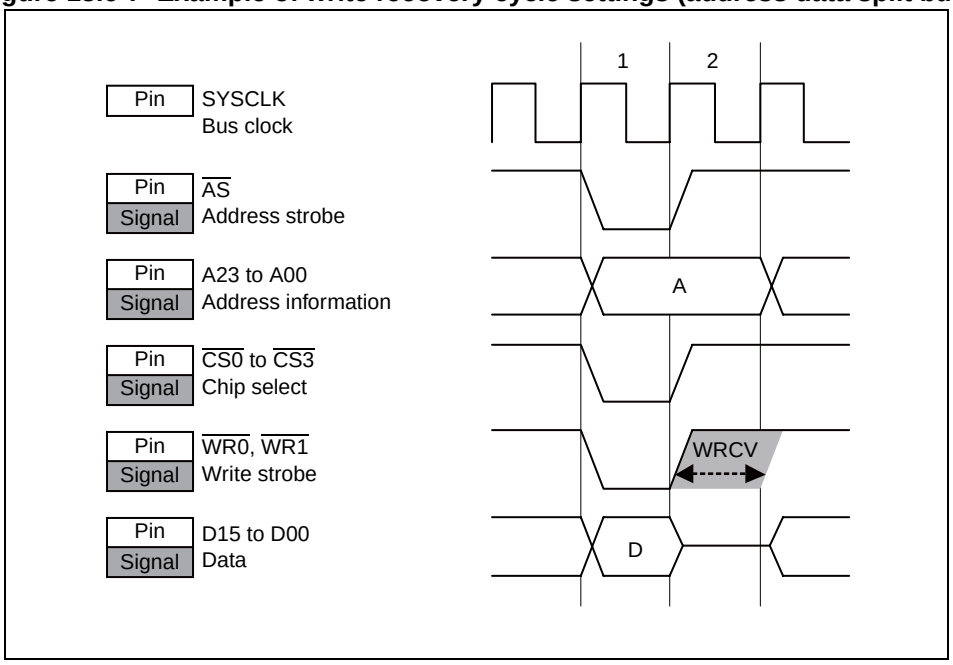


Table 13.6-11 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-7 (values other than "0" are set in the bits).

Table 13.6-11 Setting values of bits

Setting Item	Bit	Setting Value
Write recovery cycle	WRCV1, WRCV0	01

■ Address data multiplex bus

Figure 13.6-8 shows an example in which the write recovery cycle is set to 1T (T: Bus clock period).

Figure 13.6-8 Example of write recovery cycle settings (address data multiplex bus)

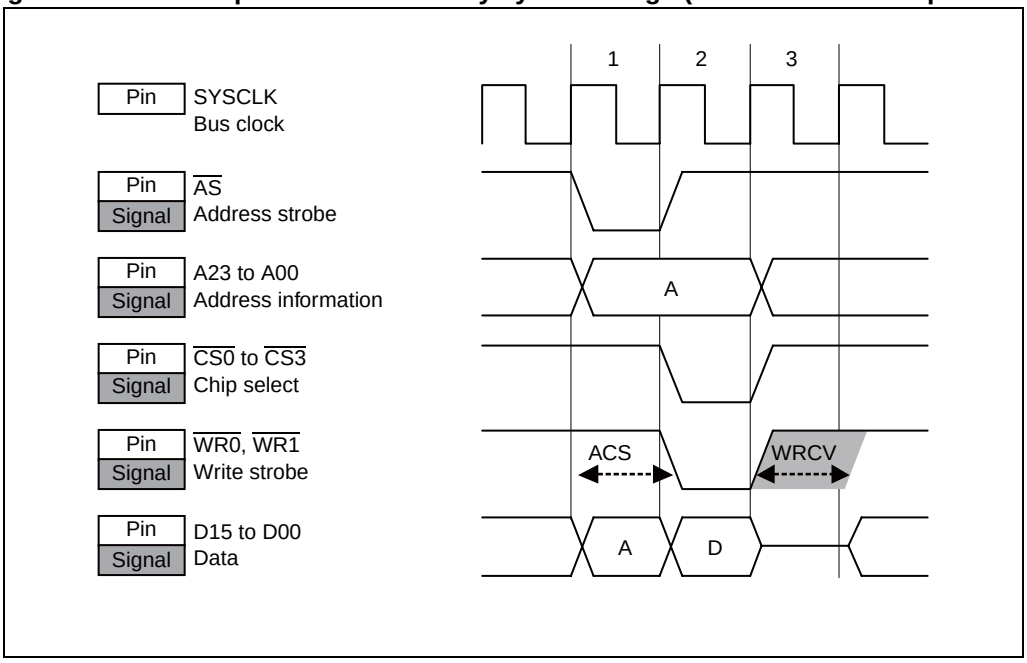


Table 13.6-12 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-8 (values other than "0" are set in the bits).

Table 13.6-12 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01
Write recovery cycle	WRCV1, WRCV0	01

Figure 13.6-8 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.5 Read Access Setup Cycle

The number of setup cycles for read access is set. The setup cycles extend the period beginning from enabling of the chip select to enabling of the read strobe.

The period beginning from enabling of the chip select to enabling of the read strobe can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the CSRD1 and CSRD0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-13 lists the setting values of the CSRD1 and CSRD0 bits in the area wait registers (AWR0 to AWR3) and the number of delay cycles.

Table 13.6-13 CSRD1 and CSRD0 bits and number of delay cycles

CSRD1	CSRD0	Number of Delay Cycles
0	0	0T (valid simultaneously with chip select)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

■ Address data split bus

Figure 13.6-9 shows an example in which the read access setup cycle is set to 1T (T: Bus clock period).

Figure 13.6-9 Example of read access setup cycle settings (address data split bus)

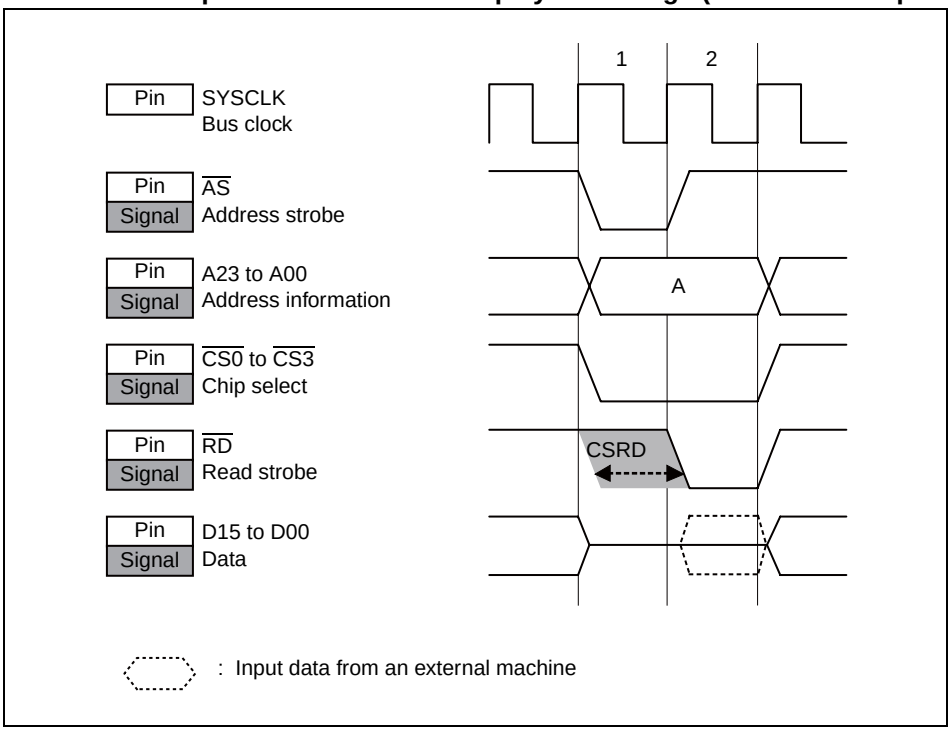


Table 13.6-14 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-9 (values other than "0" are set in the bits).

Table 13.6-14 Setting values of bits

Setting Item	Bit	Setting Value
Read access setup cycle	CSRD1, CSRD0	01

■ Address data multiplex bus

Figure 13.6-10 shows an example in which the read access setup cycle is set to 1T (T: Bus clock period).

Figure 13.6-10 Example of read access setup cycle settings (address data multiplex bus)

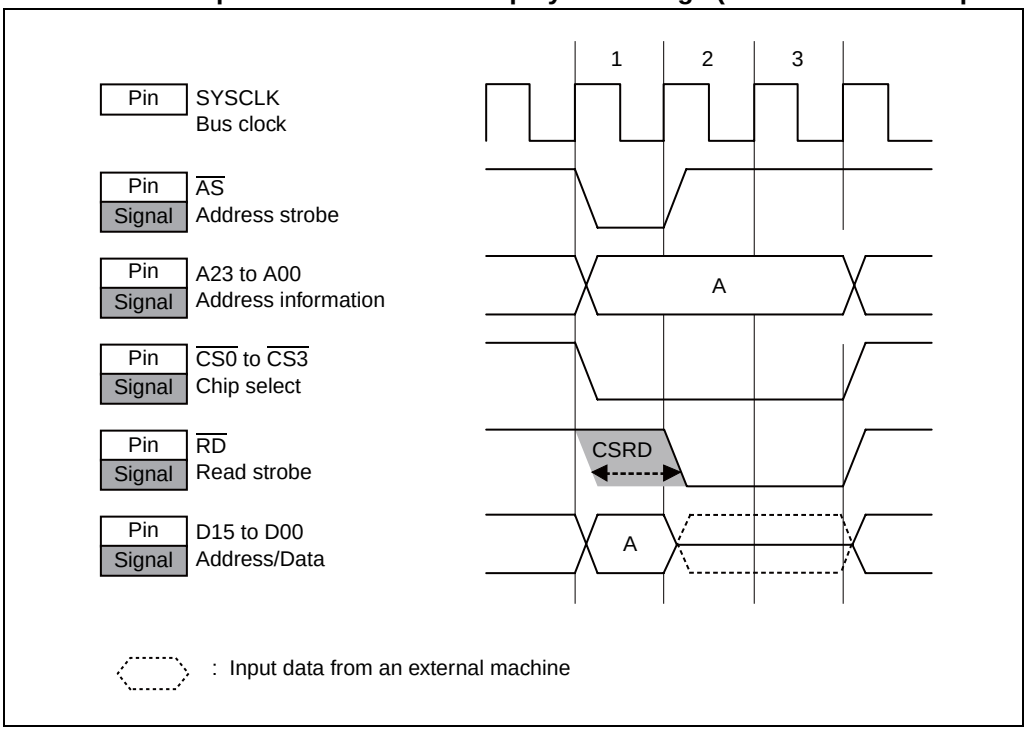


Table 13.6-15 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-10 (values other than "0" are set in the bits).

Table 13.6-15 Setting values of bits

Setting Item	Bit	Setting Value
Read access setup cycle	CSRD1, CSRD0	01

<Note>

If the address data multiplex bus is set as the bus type (ADTY=1 and BSTY=1), settings of the area wait registers (AWR0 to AWR3) must completely satisfy the following condition in conformity with the protocol:

- **ACS + CSRD ≥ 1**

  ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits

13.6.6 Read Access Hold Cycle

The number of hold cycles for read access is set. The hold cycles extend the period beginning from disabling of the chip select to disabling of the read strobe.

The period beginning from disabling of the read strobe to disabling of the chip select can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the RDCS1 and RDCS0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-16 lists the setting values of the RDCS1 and RDCS0 bits in the area wait registers (AWR0 to AWR3) and the number of delay cycles.

Table 13.6-16 RDCS1 and RDCS0 bits and number of delay cycles

RDCS1	RDCS0	Number of Delay Cycles
0	0	0T (invalid simultaneously with read strobe)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

■ Address data split bus

Figure 13.6-11 shows an example in which the read access hold cycle is set to 1T (T: Bus clock period).

Figure 13.6-11 Example of read access hold cycle settings (address data split bus)

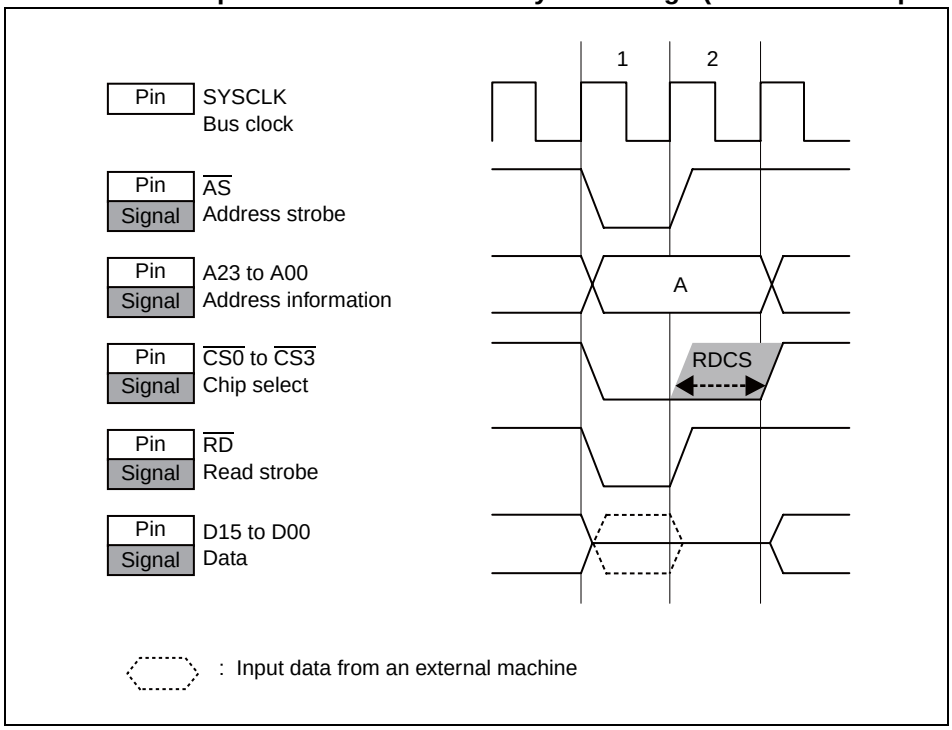


Table 13.6-17 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-11 (values other than "0" are set in the bits).

Table 13.6-17 Setting values of bits

Setting Item	Bit	Setting Value
Read access hold cycle	RDCS1, RDCS0	01

■ Address data multiplex bus

Figure 13.6-12 shows an example in which the read access hold cycle is set to 1T (T: Bus clock period).

Figure 13.6-12 Example of read access hold cycle settings (address data multiplex bus)

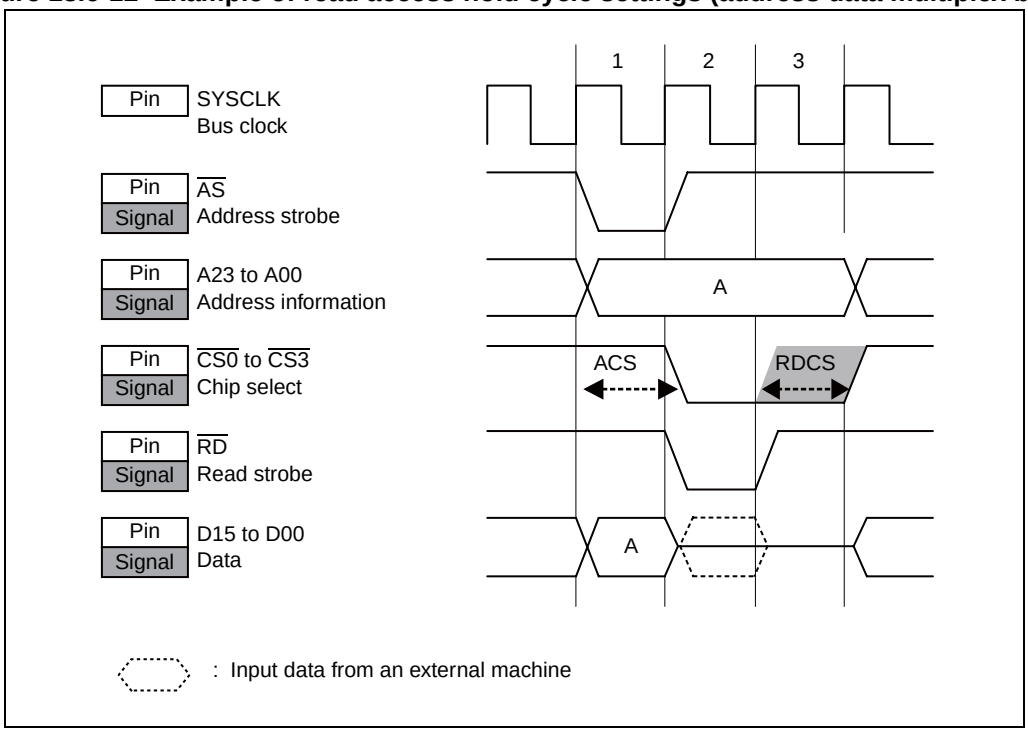


Table 13.6-18 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-12 (values other than "0" are set in the bits).

Table 13.6-18 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01
Read access hold cycle	RDCS1, RDCS0	01

Figure 13.6-12 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.7 Write Access Setup Cycle

The number of setup cycles for write access is set. The setup cycles extend the period beginning from enabling of the chip select to enabling of the write strobe.

The period beginning from enabling of the chip select to enabling of the write strobe can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the CSWR1 and CSWR0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-19 lists the setting values of the CSWR1 and CSWR0 bits in the area wait registers (AWR0 to AWR3) and the number of delay cycles.

Table 13.6-19 CSWR1 and CSWR0 bits and number of delay cycles

CSWR1	CSWR0	Number of Delay Cycles
0	0	0T (valid simultaneously with chip select)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

■ Address data split bus

Figure 13.6-13 shows an example in which the write access setup cycle is set to 1T (T: Bus clock period).

Figure 13.6-13 Example of write access setup cycle settings (address data split bus)

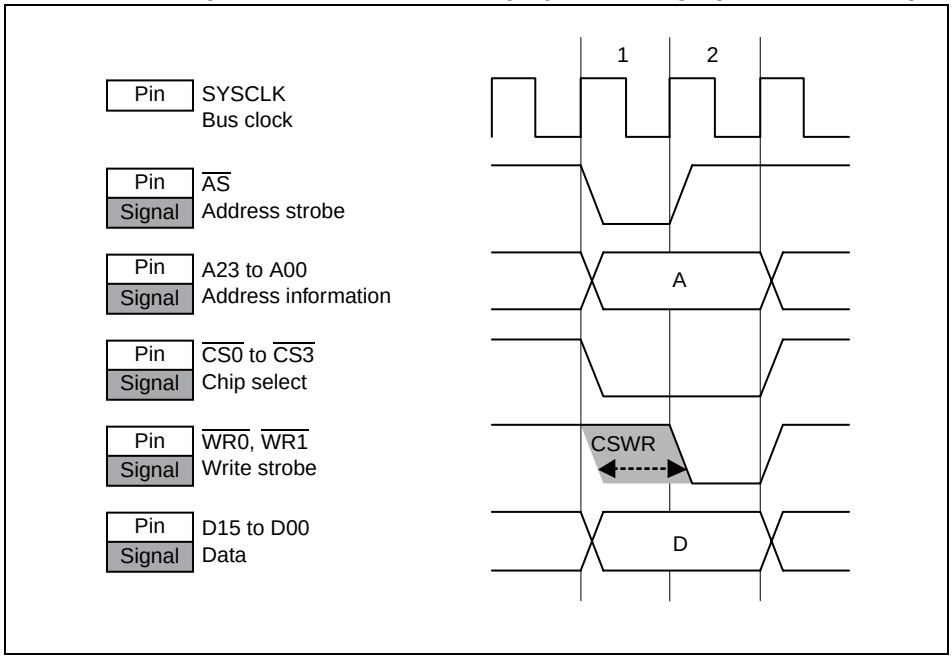


Table 13.6-20 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-13 (values other than "0" are set in the bits).

Table 13.6-20 Setting values of bits

Setting Item	Bit	Setting Value
Write access setup cycle	CSWR1, CSWR0	01

■ Address data multiplex bus

Figure 13.6-14 shows an example in which the write access setup cycle is set to 1T (T: Bus clock period).

Figure 13.6-14 Example of write access setup cycle settings (address data multiplex bus)

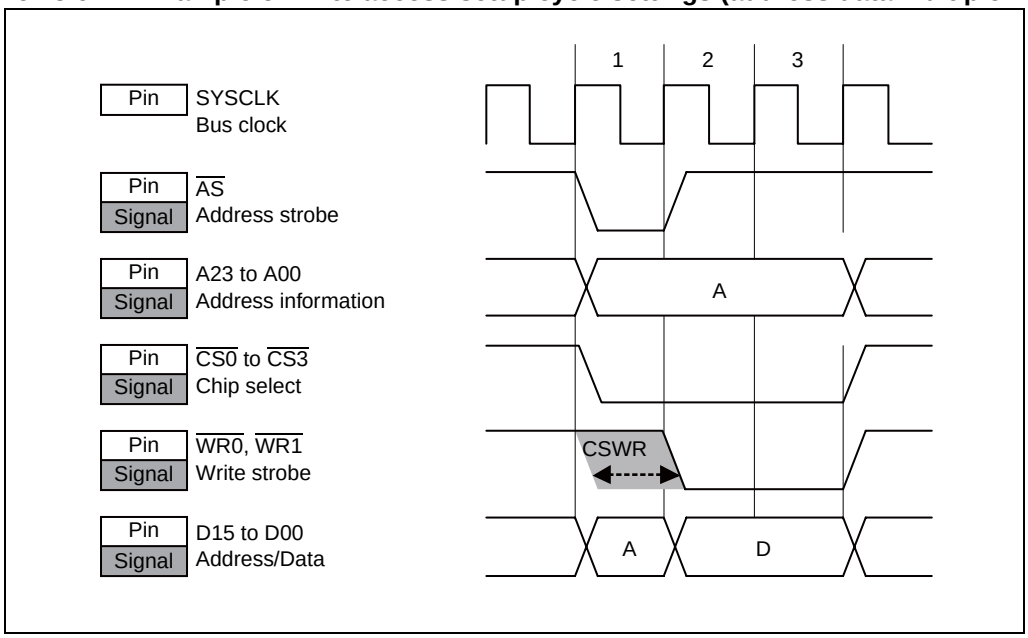


Table 13.6-21 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-13 (values other than "0" are set in the bits).

Table 13.6-21 Setting values of bits

Setting Item	Bit	Setting Value
Write access setup cycle	CSWR1, CSWR0	01

<Note>

If the address data multiplex bus is set as the bus type (ADTY=1 and BSTY=1), settings of the area wait registers (AWR0 to AWR3) must completely satisfy the following condition in conformity with the protocol:

- **ACS + CSWR ≥ 1**

  ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

13.6.8 Write Access Hold Cycle

The number of hold cycles for write access is set. The hold cycles extend the period beginning from disabling of the chip select to disabling of the write strobe.

The period beginning from disabling of the write strobe to disabling of the chip select can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the WRCS1 and WRCS0 bits on the area wait registers (AWR0 to AWR3).

Table 13.6-22 lists the setting values of the WRCS1 and WRCS0 bits in the area wait registers (AWR0 to AWR3) and the number of write access hold cycles.

Table 13.6-22 WRCS1 and WRCS0 bits and number of delay cycles

WRCS1	WRCS0	Number of Write Access Hold Cycles
0	0	0T (invalid simultaneously with write strobe)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

■ Address data split bus

Figure 13.6-15 shows an example in which the write access hold cycle is set to 1T (T: Bus clock period).

Figure 13.6-15 Example of write access hold cycle settings (address data split bus)

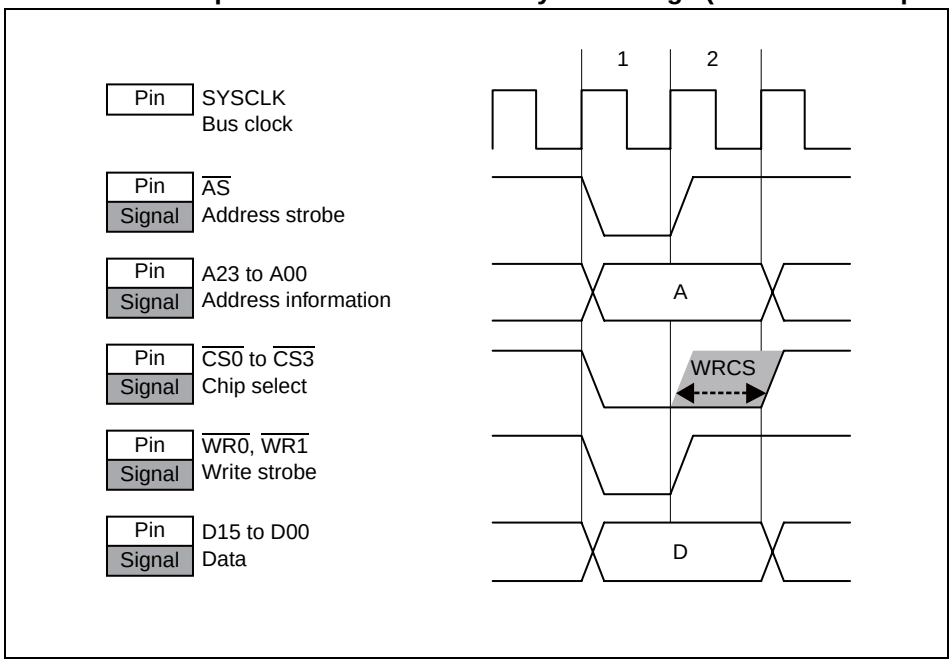


Table 13.6-23 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-15 (values other than "0" are set in the bits).

Table 13.6-23 Setting values of bits

Setting Item	Bit	Setting Value
Write access hold cycle	WRCS1, WRCS0	01

■ Address data multiplex bus

Figure 13.6-16 shows an example in which the write access hold cycle is set to 1T (T: Bus clock period).

Figure 13.6-16 Example of write access hold cycle settings (address data multiplex bus)

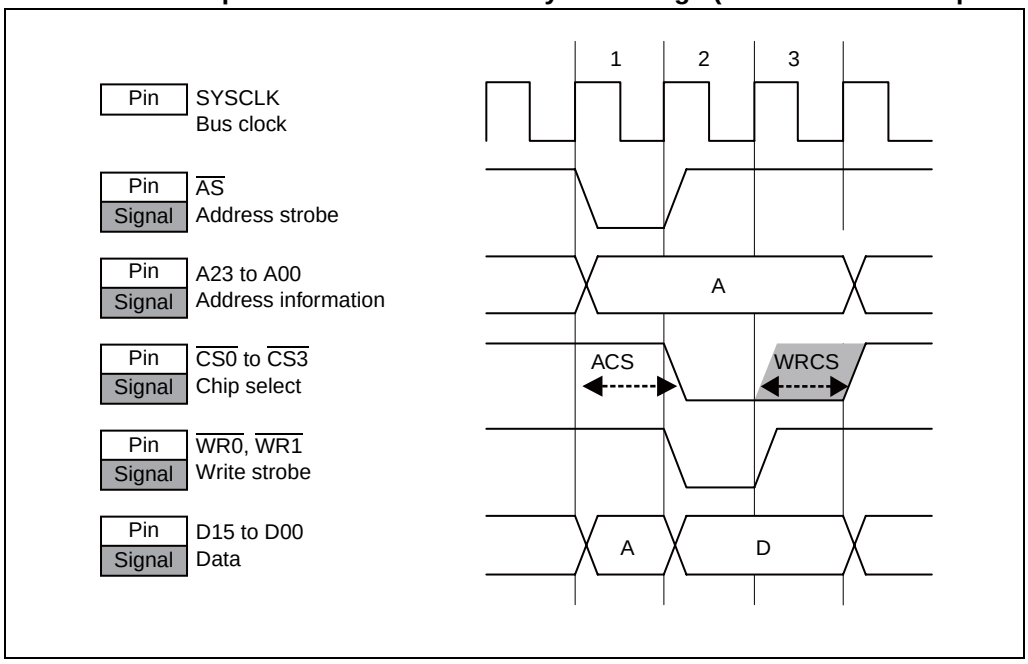


Table 13.6-24 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-16 (values other than "0" are set in the bits).

Table 13.6-24 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01
Write access hold cycle	WRCS1, WRCS0	01

Figure 13.6-16 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles because of restrictions of the address data multiplex bus protocol.

13.6.9 Chip Select Delay Cycle

The number of chip select delay cycles is set. The period from enabling of the address strobe to enabling of the chip select is set.

For chip select output, if address setup requires a certain length of time or the same CS area is accessed continuously, the address delay cycle is set to use an edge of the chip select.

The chip select enable timing following the address strobe time can be delayed by a period ranging from 0T to 3T (T: Bus clock period) with the ACS1 and ACS0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-25 lists the setting values of the ACS1 and ACS0 bits in the area wait registers (AWR0 to AWR3) and the number of delay cycles.

Table 13.6-25 ACS1 and ACS0 bits and number of delay cycles

ACS1	ACS0	Number of Delay Cycles
0	0	0T (output simultaneously with address strobe output)
0	1	1T
1	0	2T
1	1	3T

T: Bus clock period

<Note>

If the address data multiplex bus is set as the bus type (ADTY=1 and BSTY=1), settings of the area wait registers (AWR0 to AWR3) must satisfy all of the following conditions in conformity with the protocol:

- **ACS + CSRD ≥ 1**
  ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
- **ACS + CSWR ≥ 1**
  ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

■ Address data split bus

Figure 13.6-17 shows an example in which the chip select is delayed by 1T (T: Bus clock period).

Figure 13.6-17 Example of chip select delay cycle settings (address data split bus)

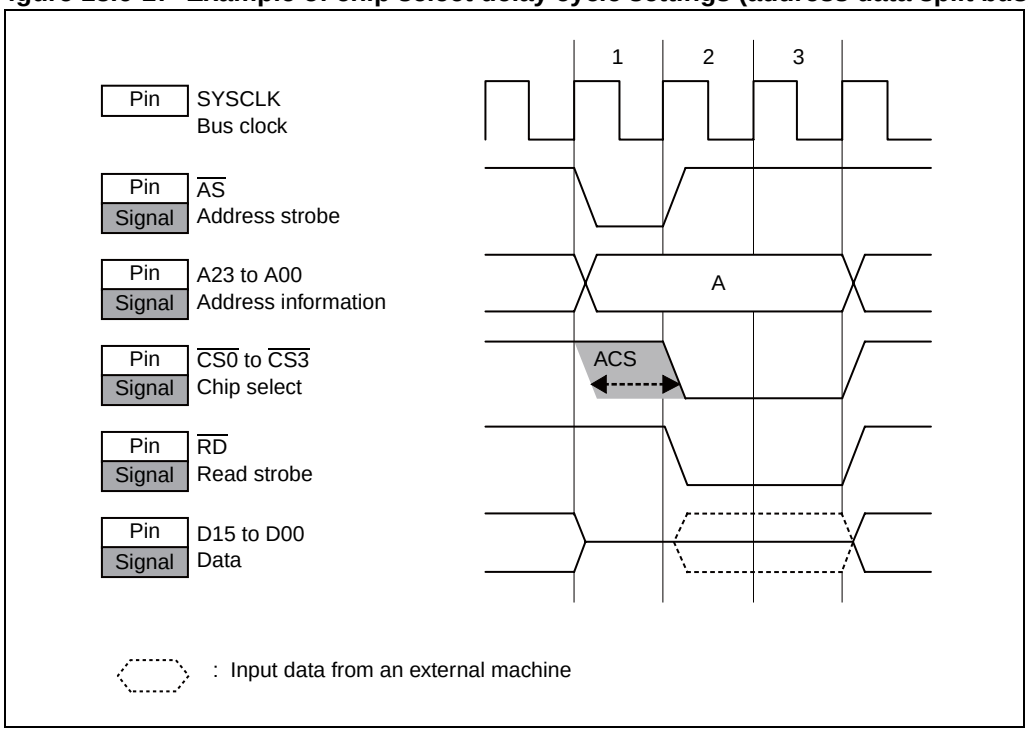


Table 13.6-26 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-17 (values other than "0" are set in the bits).

Table 13.6-26 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01

■ Address data multiplex bus

Figure 13.6-18 shows an example in which the chip select is delayed by 1T (T: Bus clock period).

Figure 13.6-18 Example of chip select delay cycle settings (address data multiplex bus)

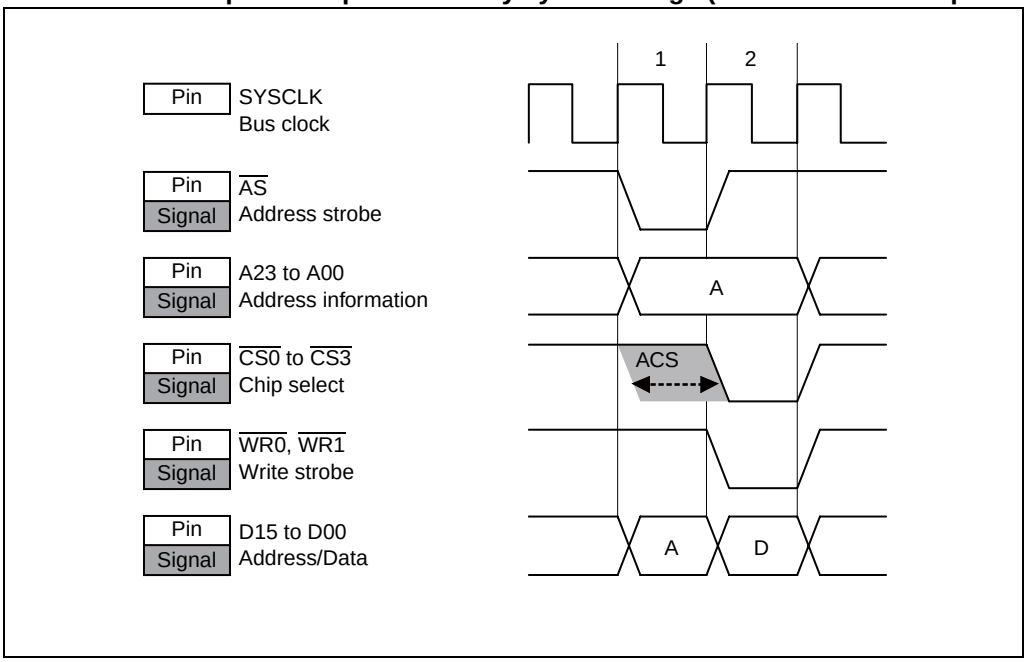


Table 13.6-27 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-18 (values other than "0" are set in the bits).

Table 13.6-27 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select delay cycles	ACS1, ACS0	01

13.6.10 Address Output Extension Cycle

The address output extension cycle for the address data multiplex bus is set(ADTY=1 and BSTY=1). It specifies the period in which address information is output to the D15 to D00 pins.

The period in which address information is output from the D15 to D00 pins (address output cycle) is at least 1T (T: bus clock cycle). The number of address output extension cycles can be specified as a period ranging from 0T to 3T (T: Bus clock period) with the ADCY1 and ADCY0 bits of the area wait registers (AWR0 to AWR3).

Table 13.6-28 lists the setting values of the ADCY1 and ADCY0 bits in the area wait registers (AWR0 to AWR3) and the address output cycle.

Table 13.6-28 ADCY1 and ADCY0 bits and extension cycle

ADCY1	ADCY0	Extension Period	Address Output Cycle (Total Length)
0	0	0T (no delay)	1T
0	1	1T	2T
1	0	2T	3T
1	1	3T	4T

T: Bus clock period

<Notes>

The period set as the address output extension cycle must be equal to or longer than that of the address strobe output extension cycle. If the set period used for the address output extension cycle is shorter than that of the address strobe output extension cycle, the address strobe output extension cycle is used in place of the address output extension cycle.

- **ADCY ≥ ASCY**
- **if (ADCY < ASCY) then ADCY = ASCY**

  ADCY: ADCY1, ADCY0 bits ASCY: ASCY1, ASCY0 bits

Even if the address output cycle is changed with the ADCY1 and ADCY0 bits of the area wait registers (AWR0 to AWR3), the output periods and output timing of the other signals are not changed.

Therefore, to change the address output cycle, make area wait register (AWR0 to AWR3) settings that satisfy all of the following conditions in conformity with the protocol:

- **ADCY + 1 ≤ ACS + CSRD**

  ADCY: ADCY1, ADCY0 bits ACS: ACS1, ACS0 bits

  CSRD: CSRD1, CSRD0 bits
- **ADCY + 1 ≤ ACS + CSWR**

  ADCY: ADCY1, ADCY0 bits ACS: ACS1, ACS0 bits

  CSWR: CSWR1, CSWR0 bits

■ Address data multiplex bus

Figure 13.6-19 shows an example in which the address output cycle is extended by 1T (T: Bus clock period).

Figure 13.6-19 Example of address output extension cycle settings

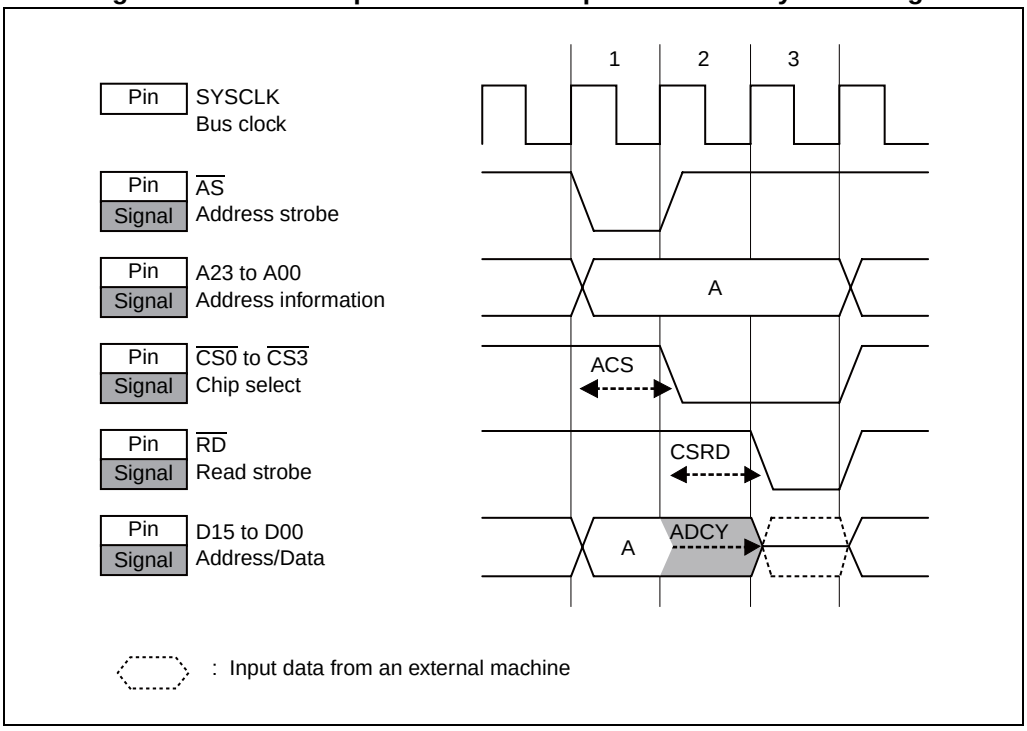


Table 13.6-29 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-19 (values other than "0" are set in the bits).

Table 13.6-29 Setting values of bits

Setting Item	Bit	Setting Value
Address output extension cycle count bits	ADCY1, ADCY0	01
Number of chip select delay cycles	ACS1, ACS0	01
Read access setup cycle	CSRD1, CSRD0	01

Table 13.6-29 shows that 1T (T: Bus clock period) is set as the number of chip select delay cycles, and 1T (T: Bus clock period) is set for the read access setup cycle. This is because of ADCY + 1 ≤ ACS + CSRD, which is a restriction of the address data multiplex bus protocol.

13.6.11 Address Strobe Output Extension Cycle

The address strobe output extension cycle is set. It specifies the period in which the address strobe is kept enabled.

The address strobe output period is at least 1T (T: bus clock cycle). The address strobe output period can be extended by 0T or 1T (T: Bus clock period) with the ASCY bit of the area wait registers (AWR0 to AWR3).

Table 13.6-30 lists the setting values of the ASCY bit in the area wait registers (AWR0 to AWR3) and the corresponding address strobe output periods.

Table 13.6-30 ASCY bit and address strobe output periods

ASCY	Extension Period	Total Output Period
0	0T (no extension)	1T
1	1T	2T

T: Bus clock period

<Note>

To extend the address strobe output period, make area wait register (AWR0 to AWR3) settings that satisfy all of the following conditions in conformity with the protocol:

- With the address data split bus as the bus type (ADTY=0 or ADTY=1 and BSTY=0)
  - **ACS + CSRD + RWT + RDCS ≥ 1**
    ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
    RWT: RWT3 to RWT0 bits RDCS: RDCS1, RDCS0 bits
  - **ACS + CSWR + WWT + WRCS ≥ 1**
    ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits
    WWT: WWT3 to WWT0 bits WRCS: WRCS1, WRCS0 bits
- With the address data multiplex bus as the bus type (ADTY=1 and BSTY=1)
  - **ACS + CSRD ≥ 2**
  - **ADCY + 1 ≤ ACS + CSRD**
    ACS: ACS1, ACS0 bits CSRD: CSRD1, CSRD0 bits
  - **ACS + CSWR ≥ 2**
  - **ADCY + 1 ≤ ACS + CSWR**
    ACS: ACS1, ACS0 bits CSWR: CSWR1, CSWR0 bits

■ Address data split bus

Figure 13.6-20 shows an example in which the address strobe output extension cycle is extended by 1T (T: Bus clock period).

Figure 13.6-20 Example of address strobe output extension cycle settings (address data split bus)

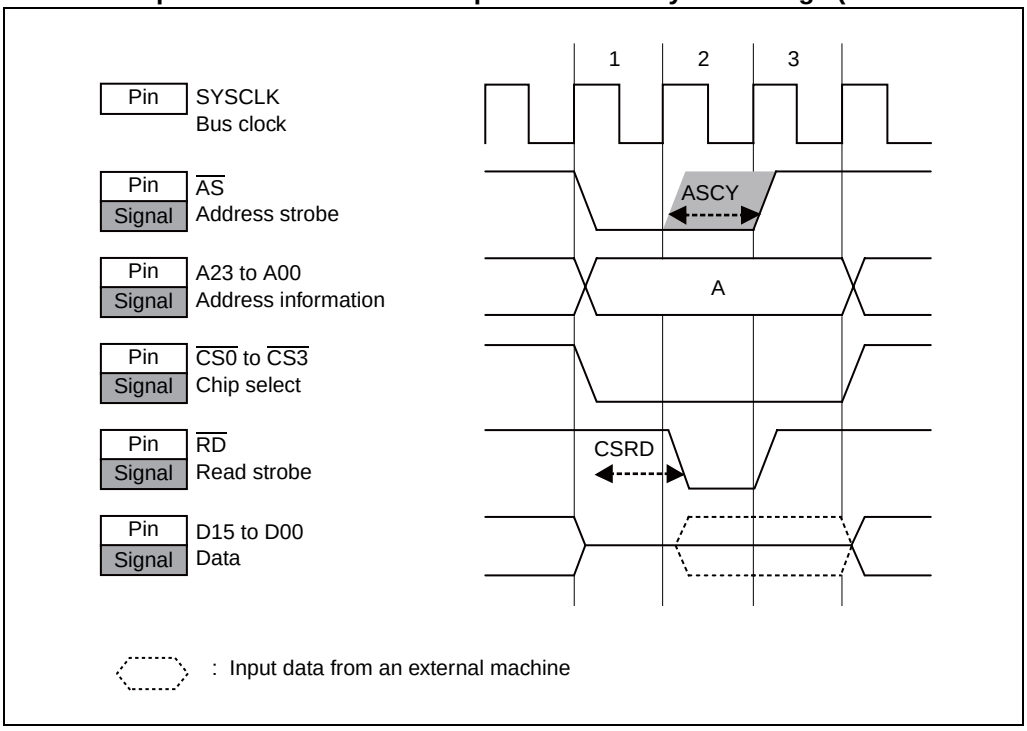


Table 13.6-31 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-20 (values other than "0" are set in the bits).

Table 13.6-31 Setting values of bits

Setting Item	Bit	Setting Value
Number of chip select strobe output extension cycles	ASCY	1
Number of read access setup cycles	CSRD1, CSRD0	01

Table 13.6-31 shows that 1T (T: Bus clock period) is set for the read access setup cycle because of protocol restrictions.

■ Address data multiplex bus

Figure 13.6-21 shows an example in which the address strobe output extension cycle is extended by 1T (T: Bus clock period).

Figure 13.6-21 Example of address strobe output extension cycle settings (address data multiplex bus)

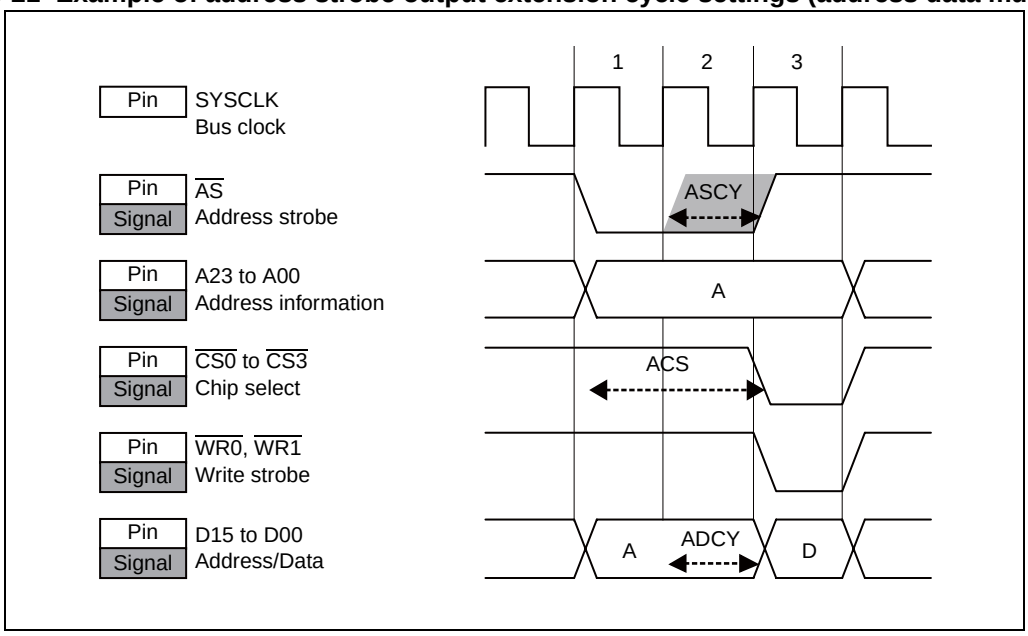


Table 13.6-32 lists the setting values of the area wait registers (AWR0 to AWR3) for the example shown in Figure 13.6-21 (values other than "0" are set in the bits).

Table 13.6-32 Setting values of bits

Setting Item	Bit	Setting Value
Number of address strobe output extension cycles	ASCY	1
Number of chip select delay cycles	ACS1, ACS0	10
Number of address output extension cycles	ADCY1, ADCY0	01

Table 13.6-32 shows that 2T (T: Bus clock period) is set as the number of address delay cycles and 1T (T: Bus clock period) is set for the address output extension cycle because of restrictions of the address data multiplex bus protocol.

13.7 Access Cycle Extension Using the RDY Pin

This section explains access cycle extension using the RDY pin of the external bus interface.

The effective period of the read strobe/write strobe can be extended by input of an "L" level signal through the RDY pin.

The read strobe/write strobe are disabled in the next cycle and the read access cycle and write access cycle are finished by input of an "H" level signal through the RDY pin.

To use the supported access cycle extension function that uses the RDY pin, write "1" to the RDYE bit in the area wait registers (AWR0 to AWR3).

<Note>

To enable this function, use the RWT3 to RWT0 bits and WWT3 to WWT0 bits of the area wait registers (AWR0 to AWR3) to specify "2" or a higher value for the read access automatic wait/write access automatic wait period.

Figure 13.7-1 shows an example of access cycle extension using the RDY pin.

Figure 13.7-1 Example of access cycle extension using the RDY pin

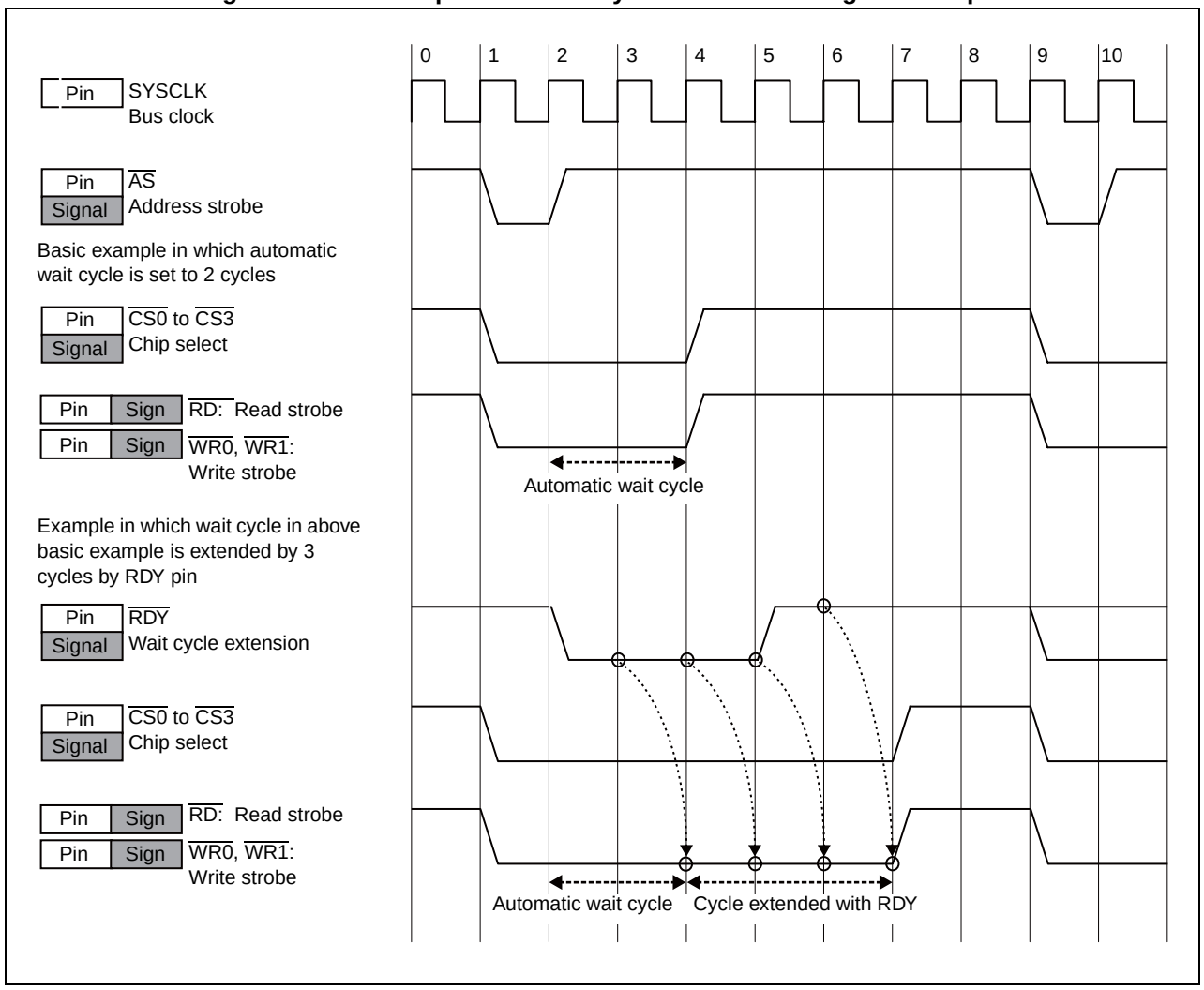


Table 13.7-1 lists the setting values of read access automatic wait registers (RWT0 to RWT3) and write access automatic wait registers (WWT0 to WWT3) for the example shown in Table 13.7-1.

Table 13.7-1 Setting values of each bit

Setting Item	Bit	Setting Value
Read access automatic wait	RWT3 to RWT0	0010
Write access automatic wait	WWT3 to WWT0	0010

<Notes>

- To not extend the automatic wait period, input an "H" level signal through the RDY pin.
- Before the start of "L" level signal input through the RDY pin, the address strobe output ("L" level output from the $\overline{AS}$ pin) and chip select output ("L" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins) must be verified.
- The "L" level signal input through the RDY pin must start before the end of the automatic wait period.
- An "H" level signal must be input through the RDY pin after the end of the required extension cycle.

13.8 Number of Access Cycles

This section explains the number of cycles required for one bus access operation of the external bus interface.

■ Address data split bus

The following formulas can be used to calculate the numbers of read access cycles and write access cycles:

- Read access

  Address/data output (1T) + ACS (0 to 3T) + CSRD (0 to 3T) + RWT (0 to 15T) + RDCS (0 to 3T) = Minimum of 1T to maximum of 25T

  ACS: Number of address output extension cycles

  CSRD: Number of read access setup cycles

  RWT: Read access automatic wait period

  RDCS: Number of read access hold cycles

  T: Bus clock period

- Write access

  Address/data output (1T) + ACS (0 to 3T) + CSWR (0 to 3T) + WWT (0 to 15T) + WRCS (0 to 3T) = Minimum of 1T to maximum of 25T

  ACS: Number of address output extension cycles

  CSWR: Number of write access setup cycles

  WWT: Write access automatic wait period

  WRCS: Number of write access hold cycles

  T: Bus clock period

■ Address data multiplex bus

The following formulas can be used to calculate the numbers of read access cycles and write access cycles:

- Read access

  Address output (1T) + ACS (0 to 3T) + CSRD (0 to 3T) + data output (1T) + RWT (0 to 15T) + RDCS (0 to 3T) = Minimum of 2T to maximum of 26T

  ACS: Number of address output extension cycles

  CSRD: Number of read access setup cycles

  RWT: Read access automatic wait period

  RDCS: Number of read access hold cycles

  T: Bus clock period

- Write access

  Address output (1T) + ACS (0 to 3T) + CSWR (0 to 3T) + data output (1T) + WWT (0 to 15T) + WRCS (0 to 3T) = Minimum of 2T to maximum of 26T

  ACS: Number of address output extension cycles

  CSWR:Number of write access setup cycles

  WWT: Write access automatic wait period

  WRCS:Number of write access hold cycles

  T: Bus clock period

13.9 Address Information and Address Alignment

This section describes address information for the external bus interface and explains address alignment.

13.9.1 Address Information

This section explains bus types and address types of the external bus interface.

Pins that output address information vary depending on the combinations of the following settings:

- Bus type (BSTY bit in an area configuration register (ACR0 to ACR3))
- Address type (ADTY bit in an area configuration register (ACR0 to ACR3))
- Data bus width (DBW1 and DBW0 bits in an area configuration register (ACR0 to ACR3))

Table 13.9-1 lists the correspondence between setting values of an area configuration register (ACR0 to ACR3) and pins that output address information.

Table 13.9-1 Correspondence between setting values of an area configuration register (ACR0 to ACR3) and pins that output address information

Address Type (ADTY)	Bus Type (BSTY)	Bus Width (DBW1, DBW0)	A23 to A00 Pins	D15 to D08 Pins (Address Output Cycle)	D7 to D00 Pins (Address Output Cycle)
0	0	00 (8 bits)	bit23 to bit0	-	-
		01 (16 bits)			
	1	00 (8 bits)	bit23 to bit0	bit7 to bit0	-
		01 (16 bits)	bit23 to bit0	bit15 to bit8	bit7 to bit0
1	0	00 (8 bits)	bit23 to bit0	-	-
		01 (16 bits)	bit24 to bit1	-	-
	1	00 (8 bits)	bit23 to bit0	bit7 to bit0	-
		01 (16 bits)	bit24 to bit1	bit16 to bit9	bit8 to bit1

ADTY : 0 = Normal output; 1 = Address shift output

BSTY : 0 = Address data split bus; 1 = Address data multiplex bus

13.9.2 Address Alignment

This section explains address alignment.

The external bus interface does not detect a misalignment of the address of an access destination. Therefore, forcible alignment applies as follows in cases of word access or half word access:

- Word access (32-bit access)

  The lower 2 bits of the address to be output are always "00" regardless of the lower 2 bits of the address specified by the program concerned.

- Half word access (16-bit access)

  If the lower 2 bits of the address specified by the program concerned are "00" or "01", the lower 2 bits of the address to be output is "00". If the lower 2 bits of the address specified by the program are "10" or "11", the lower 2 bits of the address to be output are "10".

13.10 Data Alignment

This section explains the data alignment of the external bus interface.

■ Endian

The external bus interface enables the byte ordering for the CS areas, except the CS0 area, to be set.

The byte ordering is specified by the LEDN bit in an area setting register (ASR0 to ASR3), and either big endian (LEDN = 0) or little endian (LEDN = 1) can be selected.

Big endian stores data in different ways than little endian.

Storage of **"01234567$_H$"**

- Big endian

  "01" is stored in the first byte, "23" in the second byte, "45" in the third byte, and "67" in the fourth byte.

  For details of access when big endian is set, see "13.10.1 Big Endian".

- Little endian

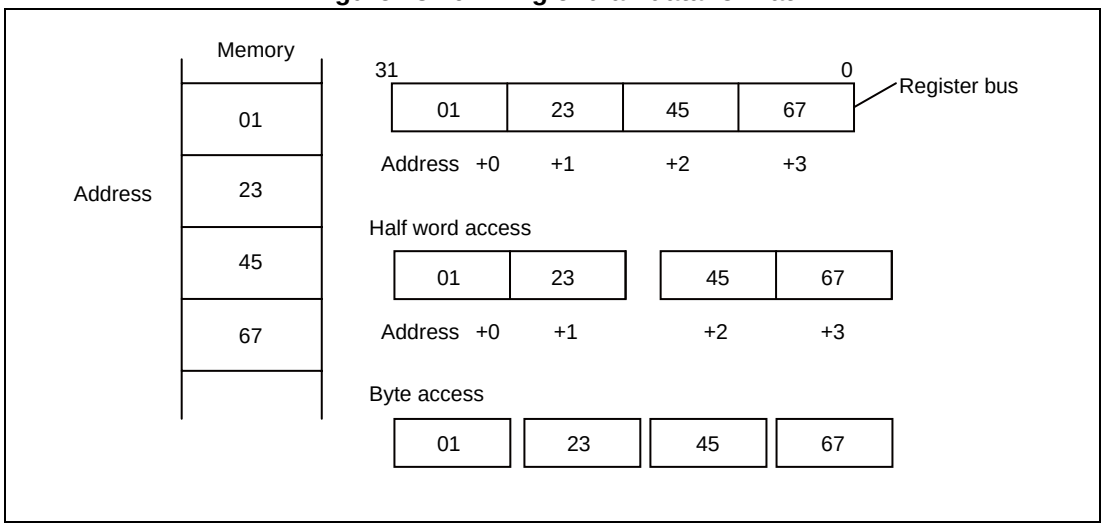
  "67" is stored in the first byte, "45" in the second byte, "23" in the third byte, and "01" in the fourth byte.

  If a data bus is connected, byte locations on the data bus are swapped according to the bus width.

  For details of access when little endian is set, see "13.10.2 Little Endian".

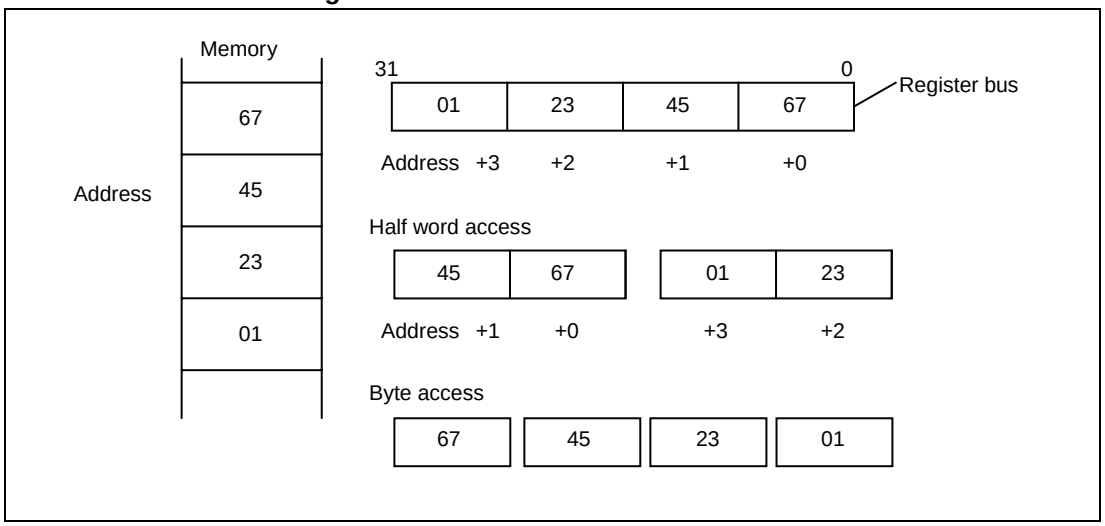
Figure 13.10-1 shows the data format of big endian, and Figure 13.10-2 shows that of little endian.

Figure 13.10-1 Big endian data format



Bytes of the word data or half word data placed in an address space are arranged in the order in which the most significant byte is located at the lowest address and the least significant byte is located at the highest address.

Figure 13.10-2 Little endian data format



Bytes of the word data or half word data placed in an address space are arranged in the order in which the most significant byte is located at the highest address and the least significant byte is located at the lowest address.

<Notes>

- The CS0 area supports only big endian. Little endian cannot be set for the CS0 area.
- If an external machine is connected, the big endian areas must physically be separated from the little endian areas.

■ Divided access

The data alignment for each access size of the external bus interface varies depending on the endian type and data bus width.

Either 8 bits or 16 bits can be selected for each CS area as the data bus width by using the DBW1 and DBW0 bits of the area configuration registers (ACR0 to ACR3).

An access operation whose access size is wider than the bus width specified by the DBW1 and DBW0 bits is performed only after being divided into multiple access operations. Table 13.10-1 lists the number of times that an access operation is divided for each access size.

Table 13.10-1 Number of divided access operations

Bus Width	Access Size		
	Byte	Half Word	Word
8 bits	1 time	2 times	4 times
16 bits	1 time	1 time	2 times

13.10.1 Big Endian

If "0" is set in the LEDN bit in an area setting register (ASR1 to ASR3), the corresponding area is treated as a big endian area.
This section explains big endian access and modes of connection.

■ Big endian access

● 16-bit external bus interface access

Table 13.10-2 lists the data alignment for each access size and corresponding control signals for the setting of big endian together with a data bus width of 16 bits.

In the case where the access size is the length of a word, access is divided into 2 access operations.

Table 13.10-2 16-bit external bus interface access

Access		Output Pin				
Size	**Address Lower 2 Bits**	**A01 to A00**	**D15 to D08**	**D07 to D00**	**$\overline{WR0}$**	**$\overline{WR1}$**
Byte	00	"00"	Data bit7 to bit0		O	
	01	"00"		Data bit7 to bit0		O
	10	"10"	Data bit7 to bit0		O	
	11	"11"		Data bit7 to bit0		O
Half word	0n	"00"	Data bit15 to bit8	Data bit7 to bit0	O	O
	1n	"10"	Data bit15 to bit8	Data bit7 to bit0	O	O
Word	nn	Divided access First time: "00"	Data bit31 to bit24	Data bit23 to bit16	O	O
		Divided access Second time: "10"	Data bit15 to bit8	Data bit7 to bit0	O	O

<Note>

In the above access examples, "0" (without shift) is set in the ADTY bit in an area configuration register (ACR0 to ACR3).

● 8-bit external bus interface access

Table 13.10-3 lists the data alignment for each access size and corresponding control signals for the setting of big endian together with a data bus width of 8 bits.

In the case where the access size is the length of a half word, access is divided into 2 access operations; in the case where the access size is the length of a word, access is divided into 4 access operations.

Table 13.10-3 8-bit external bus interface access

Access		Output Pin				
Size	**Address Lower 2 Bits**	**A01 to A00**	**D15 to D08**	**D07 to D00**	**$\overline{WR0}$**	**$\overline{WR1}$**
Byte	00	"00"	Data bit7 to bit0		O	
	01	"01"	Data bit7 to bit0		O	
	10	"10"	Data bit7 to bit0		O	
	11	"11"	Data bit7 to bit0		O	
Half word	0n	Divided access First time: "00"	Data bit15 to bit8		O	
		Divided access Second time: "01"	Data bit7 to bit0		O	
	1n	Divided access First time: "10"	Data bit15 to bit8		O	
		Divided access Second time: "11"	Data bit7 to bit0		O	
Word	nn	Divided access First time: "00"	Data bit31 to bit24		O	
		Divided access Second time: "01"	Data bit23 to bit16		O	
		Divided access Third time: "10"	Data bit15 to bit8		O	
		Divided access Fourth time: "11"	Data bit7 to bit0		O	

■ How to connect asynchronous memory

In the following connection examples, asynchronous memory is connected with external bus pins for an area for which big endian is set.

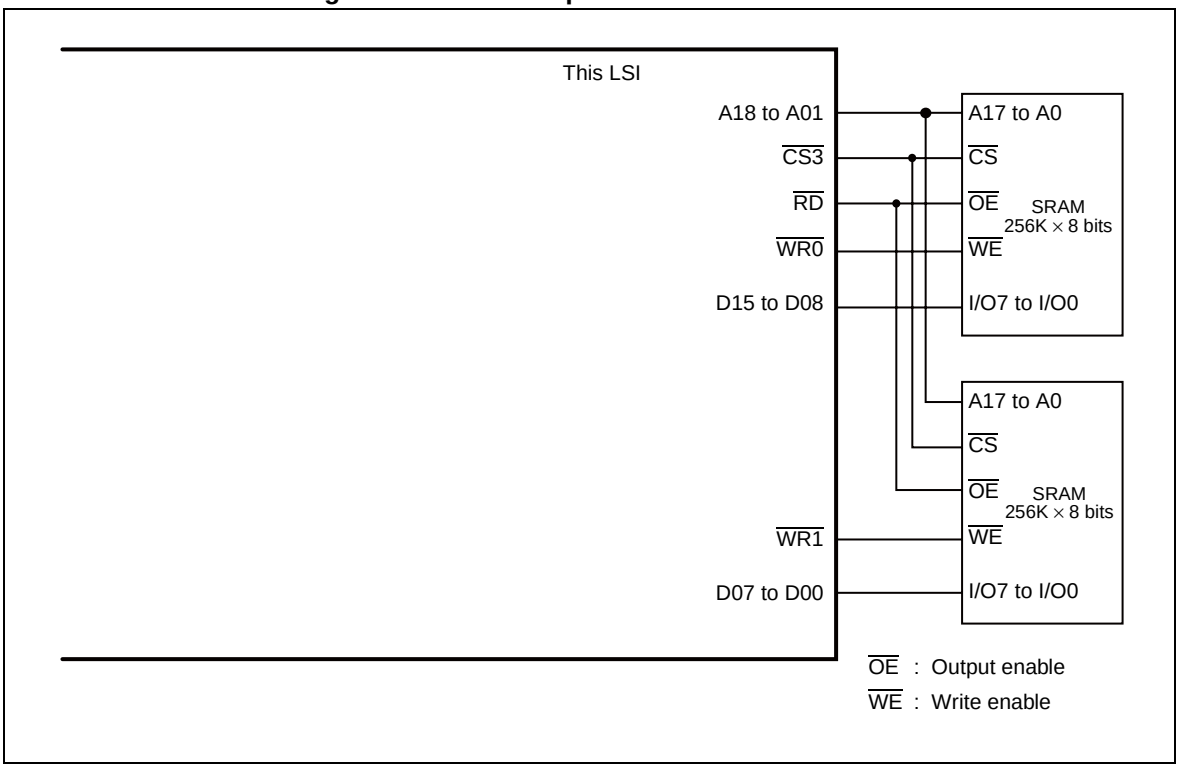
● Connection example for a 16-bit external bus interface

The CS3 area used in this example is an area with a bus width of 16 bits and the setting of big endian, and two 256K × 8-bit SRAM modules are connected to the CS3 area.

The A18 to A01 pins, $\overline{WR0}$ and $\overline{WR1}$ pins, and D15 to D00 pins are used.

Figure 13.10-3 shows the connection example with the above conditions.

Figure 13.10-3 Example for a 16-bit bus width



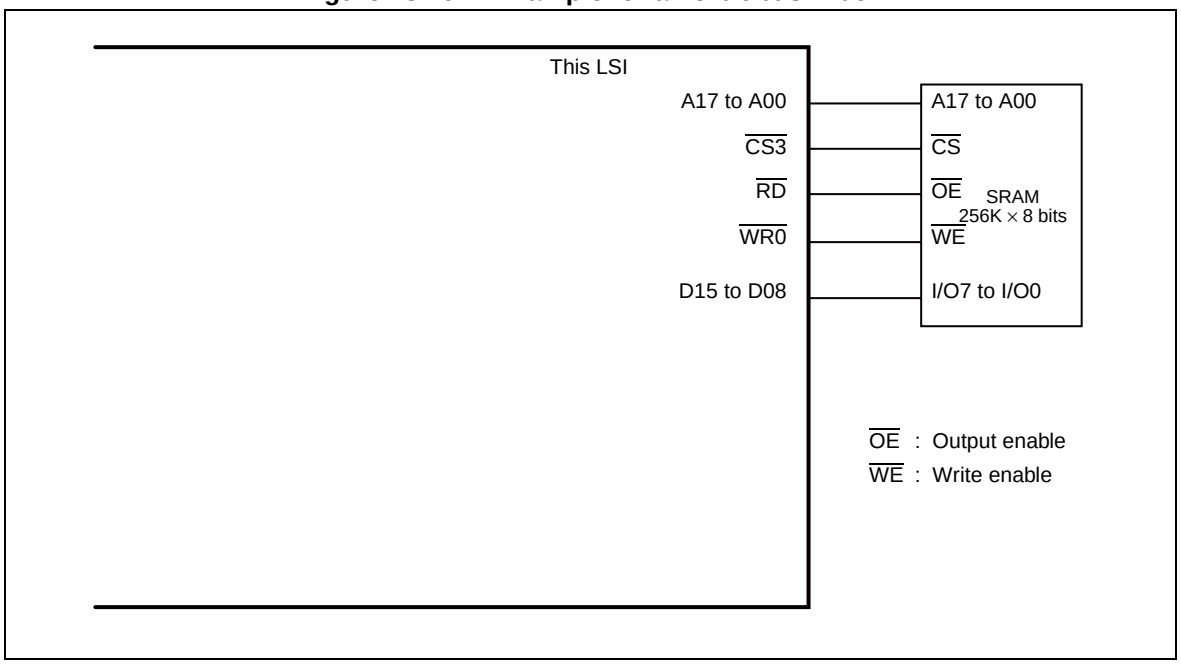
● Connection example for an 8-bit external bus interface

The CS3 area used in this example is an area with a bus width of 8 bits and the setting of big endian, and a 256K × 8-bit SRAM module is connected to the CS3 area.

The A17 to A00 pins, $\overline{WR0}$ and $\overline{WR1}$ pins, and D15 to D00 pins are used.

Figure 13.10-4 shows the connection example with the above conditions.

Figure 13.10-4 Example for an 8-bit bus width



13.10.2 Little Endian

If "1" is set in the LEDN bit in an area setting register (ASR1 to ASR3), the corresponding area is treated as a little endian area.
This section explains little endian access and modes of connection.

■ Little endian access

● 16-bit external bus interface access

Table 13.10-4 lists the data alignment for each access size and corresponding control signals for the setting of little endian together with a data bus width of 16 bits.

In the case where the access size is the length of a word, access is divided into two access operations.

Table 13.10-4 16-bit external bus interface access

Access		Output Pin				
Size	**Address Lower 2 Bits**	**A00 to A01**	**D15 to D08**	**D07 to D00**	**$\overline{WR0}$**	**$\overline{WR1}$**
Byte	00	"00"	Data bit7 to bit0		O	
	01	"01"		Data bit7 to bit0		O
	10	"10"	Data bit7 to bit0		O	
	11	"11"		Data bit7 to bit0		O
Half word	0n	"00"	Data bit7 to bit0	Data bit15 to bit8	O	O
	1n	"10"	Data bit7 to bit0	Data bit15 to bit8	O	O
Word	nn	Divided access First time: "00"	Data bit7 to bit0	Data bit15 to bit8	O	O
		Divided access Second time: "10"	Data bit23 to bit16	Data bit31 to bit24	O	O

Note: In the above access example, the ADTY bit in an area configuration register (ACR0 to ACR3) is set to "0" (without shift).

● 8-bit external bus interface access

Table 13.10-5 lists the data alignment for each access size and corresponding control signals for the setting of little endian together with a data bus width of 8 bits.

In the case where the access size is the length of a half word, access is divided into 2 access operations; in the case where the access size is the length of a word, access is divided into 4 access operations.

Table 13.10-5 8-bit external bus interface access

Access		Output Pin				
Size	**Address Lower 2 Bits**	**A00 to A01**	**D15 to D08**	**D07 to D00**	**$\overline{WR0}$**	**$\overline{WR1}$**
Byte	00	"00"	Data bit7 to bit0		O	
	01	"01"	Data bit7 to bit0		O	
	10	"10"	Data bit7 to bit0		O	
	11	"11"	Data bit7 to bit0		O	
Half word	0n	Divided access First time: "00"	Data bit7 to bit0		O	
		Divided access Second time: "01"	Data bit15 to bit8		O	
	1n	Divided access First time: "10"	Data bit7 to bit0		O	
		Divided access Second time: "11"	Data bit15 to bit8		O	
Word	nn	Divided access First time: "00"	Data bit7 to bit0		O	
		Divided access Second time: "01"	Data bit15 to bit8		O	
		Divided access Third time: "10"	Data bit23 to bit16		O	
		Divided access Fourth time: "11"	Data bit31 to bit24		O	

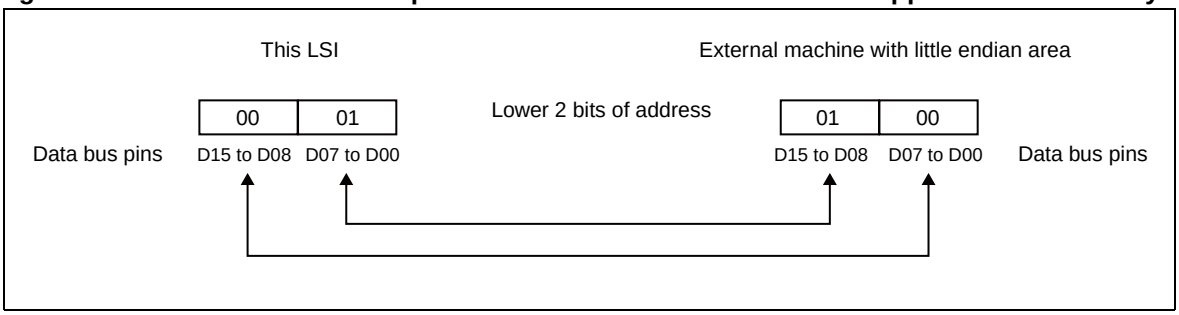
■ How to connect an external machine with a little endian area

The following figures show how to connect data bus and byte enable signals with a little endian external machine.

● Connection example for a 16-bit external bus interface

If an external machine with a little endian area with a bus width of 16 bits is connected, data bus width must be swapped in the unit of byte as shown in Figure 13.10-5.

Figure 13.10-5 Connection example in which the data bus width is swapped in the unit of byte

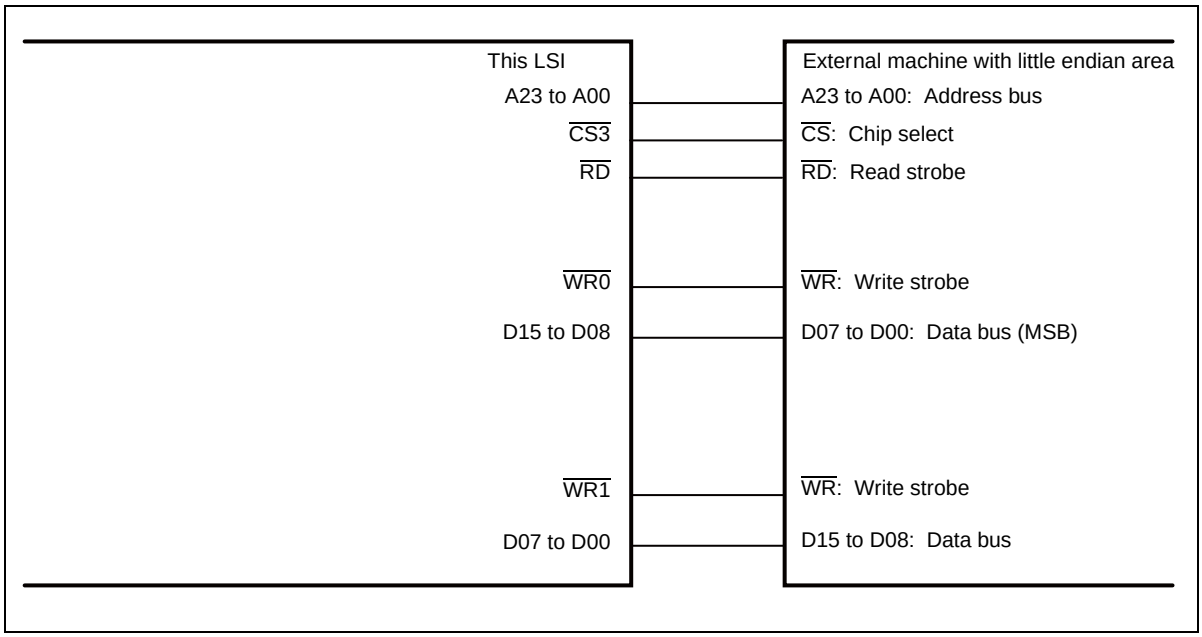


In the above example, the external machine with the little endian area is connected to the CS3 area. The CS3 area is set with a 16-bit bus width and as a little endian area.

Figure 13.10-6 shows the connection example with the above conditions.

The $\overline{WR0}$ and $\overline{WR1}$ pins and the D15 to D00 pins are used.

Figure 13.10-6 How to connect an external machine with a little endian area with a data bus when the bus width is 16 bits



● Connection example for an 8-bit external bus interface

Figure 13.10-7 shows the bit positions of the data bus and byte enable signal used in the connection example.

Figure 13.10-7 Bit positions of the data bus and byte enable signal

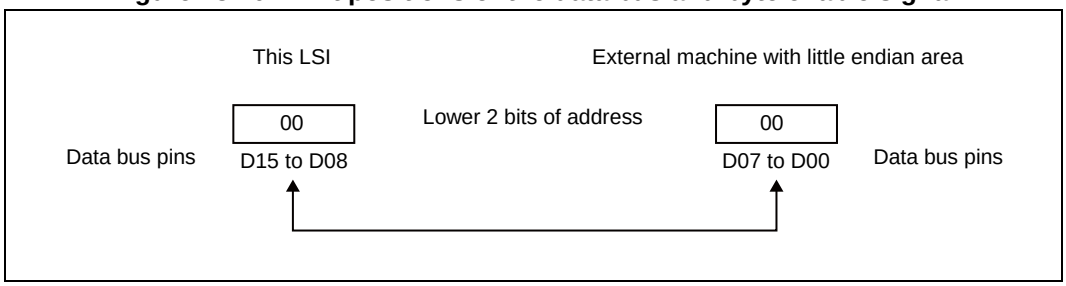
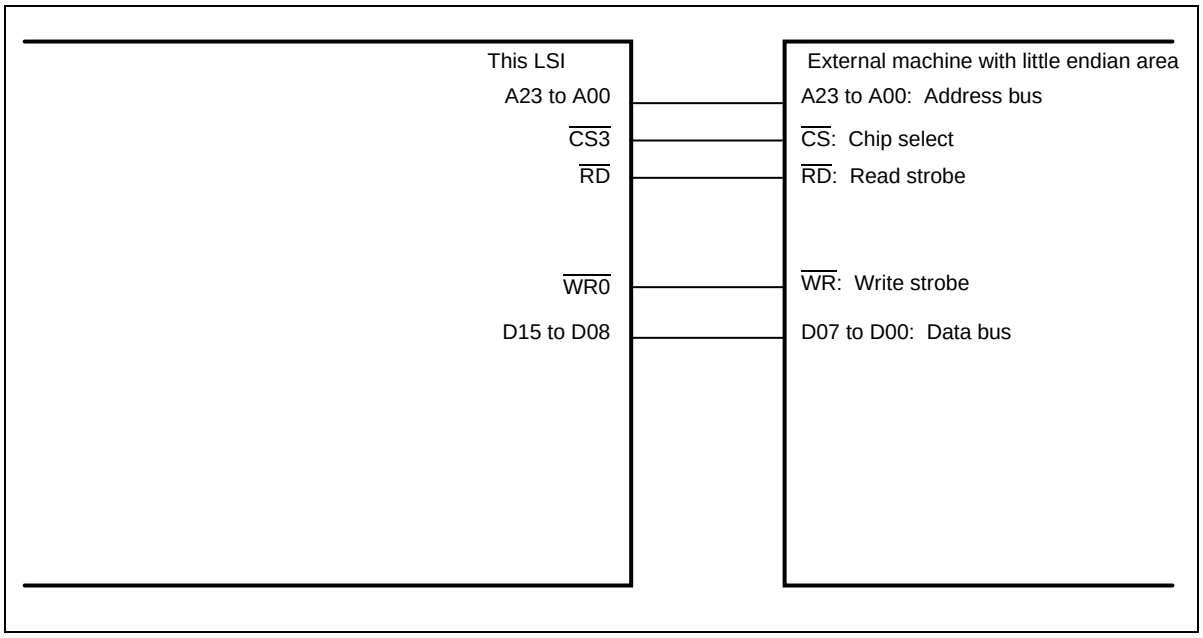


Figure 13.10-8 shows that an external machine with a little endian area is connected to the CS3 area. The CS3 area is set with a 8-bit bus width and as a little endian area.

Figure 13.10-8 shows the connection example with the above conditions.

The $\overline{WR0}$ pin and the D15 to D08 pins are used.

Figure 13.10-8 How to connect an external machine with a little endian area with a data bus when the bus width is 8 bits



13.11 External Bus DMA Transfer

This chapter explains external bus DMA transfer with external bus interface.

■ Overview

A DMA transfer with an external bus area is activated by input of a transfer request to the DREQ0 to DREQ3 pins.

- The following 3 types of pins concerning DMA transfer are provided:
  - DREQ0 to DREQ3 pins

    These pins are used to input transfer requests. A DMA transfer is activated when an effective level/edge is detected on these pins.
  - DACK0 to DACK3 pins

    A transfer request acceptance signal is output to these pins when the DMA controller (DMAC) accepts a transfer request.
  - DEOP0 to DEOP3 pins

    The DMA controller (DMAC) outputs a transfer end signal to these pins when a DMA transfer ends.
- The decision on whether to detect the level or edge of a transfer request signal is automatically made according to the DMA transfer type (demand transfer or block/burst transfer).
- Depending on the transfer type, either of the following can be selected for the detection target level/ edge in a transfer request signal:
  - For demand transfer: "H" level/"L" level
  - For block transfer and burst transfer: Rising edge/falling edge

  If a DMA transfer is activated by input of a transfer request from the DREQ0 to DREQ3 pins, the transfer source , the transfer destination, or both of them must be external bus areas.
- Either the "L" level or "H" level can be selected as the effective level of transfer request acceptance signals and transfer end signals.
- Transfer request acceptance signals and transfer end signals are output in sync with DMA transfer access to external bus areas.

13.11.1 Transfer Requests with the DREQ0 to DREQ3 Pins

This section explains the DREQ0 to DREQ3 pins, which are used to input DMA transfer requests.

■ Overview

A DMA transfer request is issued when the specified level/edge is input from the DREQ0 to DREQ3 pins to a channel for which the DREQ0 to DREQ3 pins are specified as the transfer request generation source by the RS1 and RS0 bits (RS1, RS = 1) in a DMA channel control register (DCCR0 to DCCR3). The level/edge is specified by the REQL bit in a DMA transfer register (DMAR0 to DMAR3).

For details of the setting of the transfer request generation source, see "33.4.5 DMA Channel Control Registers (DCCR0 to DCCR7)".

● Detection target level/edge

The transfer request detection level/edge can be set for each channel by using the REQL bit of the DMA transfer registers (DMAR0 to DMAR3). However, whether to use level detection or edge detection is determined depending on the transfer mode specified by the TM1 and TM0 bits in a DMA channel control register (DCCR0 to DCCR3) of the DMA controller (DMAC).

Table 13.11-1 lists the detection target level/edge of the DREQ0 to DREQ3 pins.

Table 13.11-1 Detection target level/edge of the DRQ0 to DRQ3 pins

Transfer Mode of DMA Controller (DMAC)		REQL
Demand Transfer Time	Block Transfer/Burst Transfer Time	
"L" level	Falling edge	0
"H" level	Rising edge	1

- Block transfer

  In this transfer mode, 1 block of data is transferred when a transfer request is generated. The next block of data is transferred when another transfer request is detected after the transfer of that first block of data.

- Burst transfer

  In this transfer mode, the DMA controller (DMAC) transfers data when a transfer request is generated, and it continues transferring it block by block until all the data has been transferred.

- Demand transfer

  In this transfer mode, the DMA controller (DMAC) starts a data transfer when a transfer request is generated. The data transfer continues until the transfer is completed or the transfer request is canceled.

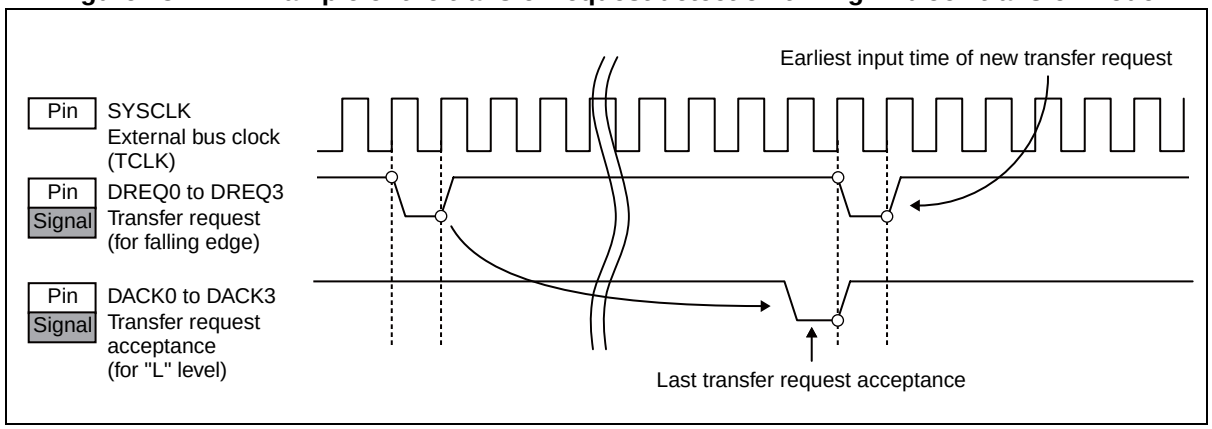
■ Operation in block transfer mode

When the edge specified by the REQL bit in a DMA transfer register (DMAR0 to DMAR3) is detected by means of an input signal from the DREQ0 to DREQ3 pins, a transfer request of the appropriate channel is issued to the DMA controller (DMAC).

Transfer operations of 1 block are performed with one transfer request.

Figure 13.11-1 shows an example of the transfer request detection timing in block transfer mode.

Figure 13.11-1 Example of the transfer request detection timing in block transfer mode



For one transfer request, the DMA controller (DMAC) outputs transfer request acceptance signals of 1 block size from the DACK0 to DACK3 pins. For the last transfer, the DMAC outputs 1 transfer end signal from the DEOP0 to DEOP3 pins at the same time.

However, if the size of data to be transferred from the DMA controller (DMAC) exceeds the bus width of the external bus interface, the access operation from the DMA controller (DMAC) is divided into as many operations as necessary before the data is transferred.

Therefore, the transfer request acceptance signal or transfer end signal output by the DMA controller (DMAC) is added to all of the divided access operations.

- **Example: Transfer count=2, block size = 3, transfer size = 32 bits, external bus width = 8 bits**

  $1 \times 3 \times (32 / 8) = 12$ transfer acceptance signals and $1 \times (32 / 8) = 4$ transfer end signals are output for one transfer request.

<Note>

Before issuing a new transfer request to the same channel, verify the last transfer request acceptance signal corresponding to the previous transfer request.

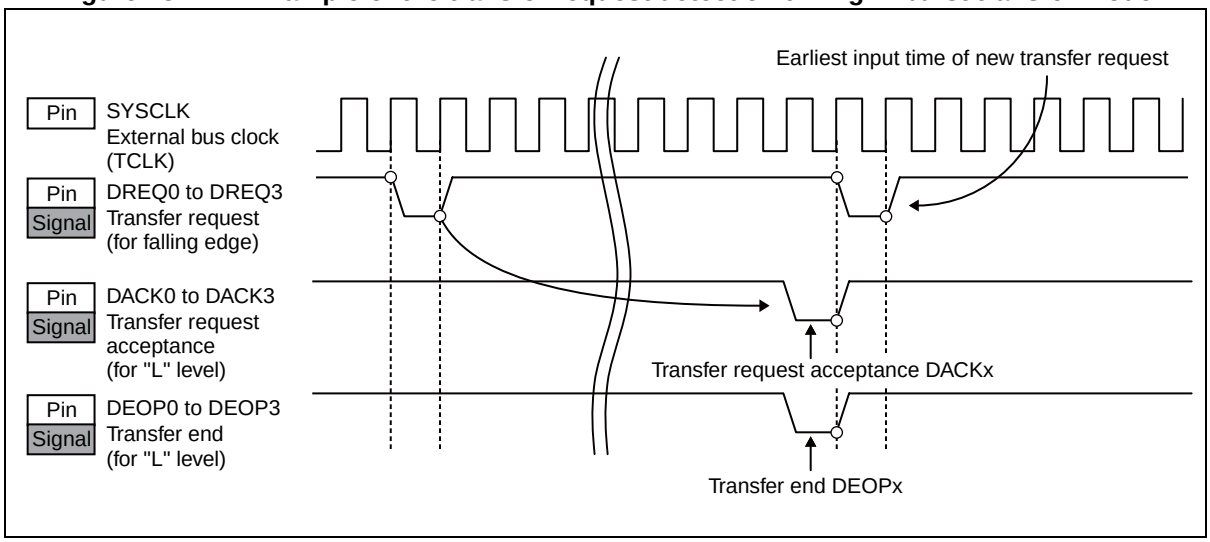
■ Operation in burst transfer mode

When the edge specified by the REQL bit in a DMA transfer register (DMAR0 to DMAR3) is detected by means of an input signal from the DREQ0 to DREQ3 pins, a transfer request of the appropriate channel is issued to the DMA controller (DMAC).

1 block multiplied by the number of transfers are performed with one transfer request.

Figure 13.11-2 shows an example of the transfer request detection timing in burst transfer mode.

Figure 13.11-2 Example of the transfer request detection timing in burst transfer mode



For one transfer request, the DMA controller (DMAC) outputs 1 block size multiplied by transfer request acceptance signals of the number of transfers from the DACK0 to DACK3 pins. For the last transfer, the DMAC outputs 1 transfer end signal from the DEOP0 to DEOP3 pins at the same time.

However, if the size of data to be transferred from the DMA controller (DMAC) exceeds the bus width of the external bus interface, the access operation from the DMA controller (DMAC) is divided into as many operations as necessary before the data is transferred.

Therefore, the transfer request acceptance signal or transfer end signal output by the DMA controller (DMAC) is added to all of the divided access operations.

Example: Transfer count = 2, transfer block size = 3, DMA transfer size = 32 bits, external bus width = 8 bits

$2 \times 3 \times (32 / 8) = 24$ transfer request acceptance signals and $1 \times (32 / 8) = 4$ transfer end signals are output for one transfer request.

<Note>

Before issuing a new transfer request to the same channel, verify the last transfer request acceptance signal/transfer end signal corresponding to the previous transfer request.

■ Operation in demand transfer mode

When a signal with the level specified by the REQL bit in a DMA transfer register (DMAR0 to DMAR3) is input to the DREQ0 to DREQ3 pins, a transfer request of the appropriate channel is issued to the DMA controller (DMAC).

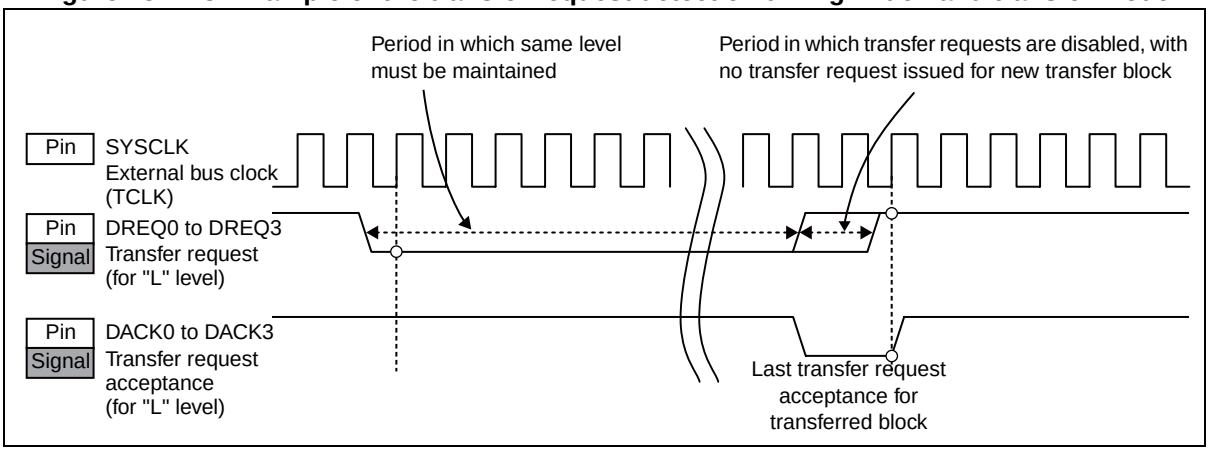
Once a transfer request is generated, the data of 1 block unit is transferred continuously while signals at the effective level are input from the DREQ0 to DREQ3 pins.

If the next transfer operation (transfer operations of 1 block) is not desired after transfer operations of 1 block have been completed, take the following steps. First, set the signal on the DREQ0 to DREQ3 pins to the invalid level while the last transfer request acceptance signal for the block being transferred is output from the DACK0 to DACK3 pins.

If the transfer stop signal (invalid level of signals from DREQ0 to DREQ3 pins) is input after the above time, it will be recognized as the next transfer request signal, which may cause a DMA transfer overrun. Lastly, taking into consideration the feasibility of an external generation circuit for the DREQ0 to DREQ3 pins, set the output period of the transfer acceptance signal from the DACK0 to DACK3 pins to be at least 2T (T: Bus clock period).

Figure 13.11-3 shows an example of the transfer request detection timing in demand transfer mode.

Figure 13.11-3 Example of the transfer request detection timing in demand transfer mode


Period in which same level must be maintained
Period in which transfer requests are disabled, with no transfer request issued for new transfer block
Pin
SYSCLK
External bus clock (TCLK)
Pin
DREQ0 to DREQ3
Signal
Transfer request (for "L" level)
Pin
DACK0 to DACK3
Signal
Transfer request acceptance (for "L" level)
Last transfer request acceptance for transferred block

For one transfer request, the DMA controller (DMAC) outputs transfer request acceptance signals of 1 block size from the DACK0 to DACK3 pins. For the last transfer, the DMAC outputs 1 transfer end signal from the DEOP0 to DEOP3 pins at the same time.

However, if the size of data to be transferred from the DMA controller (DMAC) exceeds the bus width of the external bus interface, the access operation from the DMA controller (DMAC) is divided into as many operations as necessary before the data is transferred.

Therefore, the transfer request acceptance signal or transfer end signal output by the DMA controller (DMAC) is added to all of the divided access operations.

- **Example: Transfer count=2, block size = 3, transfer size = 32 bits, external bus width = 8 bits**

  $1 \times 3 \times (32 / 8) = 12$ transfer request acceptance signals and $1 \times (32 / 8) = 4$ transfer end signals are output for one transfer request.

13.11.2 Transfer Request Acceptance with the DACK0 to DACK3 Pins

This section explains the DACK0 to DACK3 pins, which output DMA transfer request acceptance signals.

■ Transfer request acceptance

Transfer request acceptance signals are output from the DACK0 to DACK3 pins in sync with access to the external bus interface only if either the DMA transfer source type or DMA transfer destination type is set to be other than memory in the settings of the ST bit and DT bit (ST, DT = 1)in a DMA channel control register (DCCR0 to DCCR3) of the DMA controller (DMAC).

■ Output level setting

The ACKL bit in a DMA transfer register (DMAR0 to DMAR3) specifies which level of a signal input from the DACK0 to DACK3 pins indicates acceptance of a transfer request.

Explanation	ACKL
"L" level	0
"H" level	1

■ Output timing selection

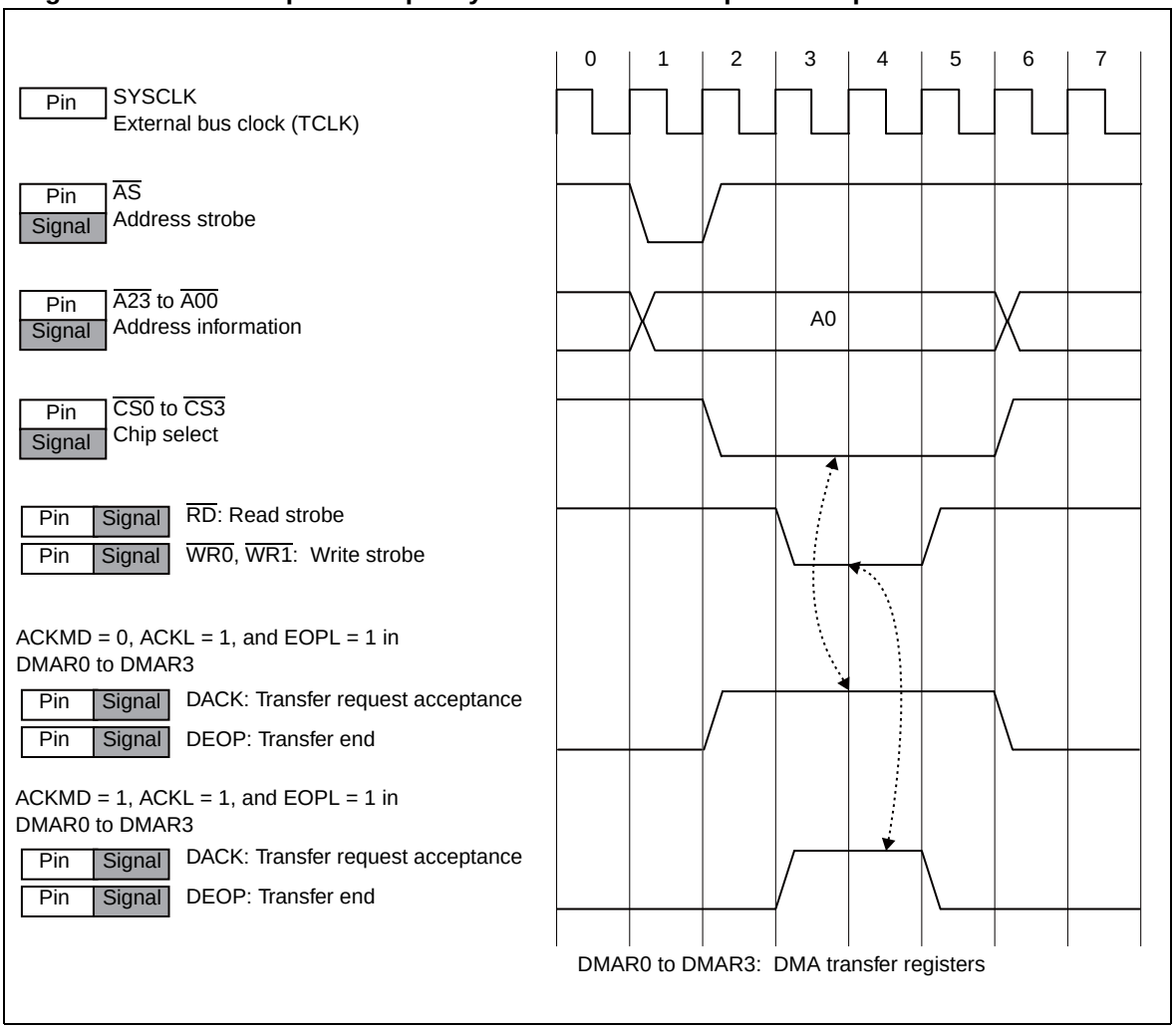
The DMA controller (DMAC) outputs a transfer request acceptance signal from the DACK0 to DACK3 pins when it accepts a transfer request from the external bus interface.

The ACKMD bit of the DMA transfer registers (DMAR0 to DMAR3) is used to set the timing of such signal output by the DMA controller (DMAC).

Explanation	ACKMD
Output together with the chip select ("L" level output from the $\overline{CS0}$ to $\overline{CS3}$ pins)	0
Output together with the read strobe/write strobe ("L" level output from the $\overline{RD}$ pin/ $\overline{WR0}$, $\overline{WR1}$pins)	1

Figure 13.11-4 shows an example of output cycles for transfer request acceptance and transfer end.

Figure 13.11-4 Example of output cycles for transfer request acceptance and transfer end



If the size of data to be transferred from the DMA controller (DMAC) exceeds the bus width of the external bus interface, the access operation from the DMA controller (DMAC) is divided into as many operations as necessary before the data is transferred.

Therefore, the transfer request acceptance signal output by the DMA controller (DMAC) is added to all of the divided access operations.

13.11.3 Transfer End Signal by the DEOP0 to DEOP3 Pins

This section explains the DEOP0 to DEOP3 pins, which output transfer end signals.

■ Transfer end signal

Transfer end signals are output from the DEOP0 to DEOP3 pins in sync with access to the external bus interface when the external bus interface access is the last DMA transfer only if either the DMA transfer source type or DMA transfer destination type is set to be other than memory in the settings of the ST bit and DT bit (ST, DT = 1) in a DMA channel control register (DCCR0 to DCCR3) of the DMA controller (DMAC).

■ Output level setting

Once the DMA controller (DMAC) transfers the last of the data to be transferred, the level of signal output from the DEOP0 to DEOP3 pins is changed, which indicates the end of the transfer.

The EOPL bit of the DMA transfer registers (DMAR0 to DMAR3) can be specified for each channel. With this specification, the end of a DMA transfer is indicated by a specific level of signal input from the DEOP0 to DEOP3 pins to the external bus interface.

Explanation	EOPL
"L" level	0
"H" level	1

■ Output timing selection

Each transfer end signal is output with the same timing and in the same cycle as a transfer request acceptance signal output from the DACK0 to DACK3 pin. The transfer end signal is output once during the last transfer access.

If the size of data to be transferred from the DMA controller (DMAC) exceeds the bus width of the external bus interface, the access operation from the DMA controller (DMAC) is divided into as many operations as necessary before the data is transferred.

Therefore, the transfer end signal output by the DMA controller (DMAC) is added to all of the divided access operations.

13.12 CS Area Setting Procedure

This section explains how to set the CS area.

Note the following about making CS area settings:

- CS area setting must be made at the initial setting time after a reset and must not be changed at a later time.
- To make settings or changes for CS areas, use the initial setting program stored in ROM.

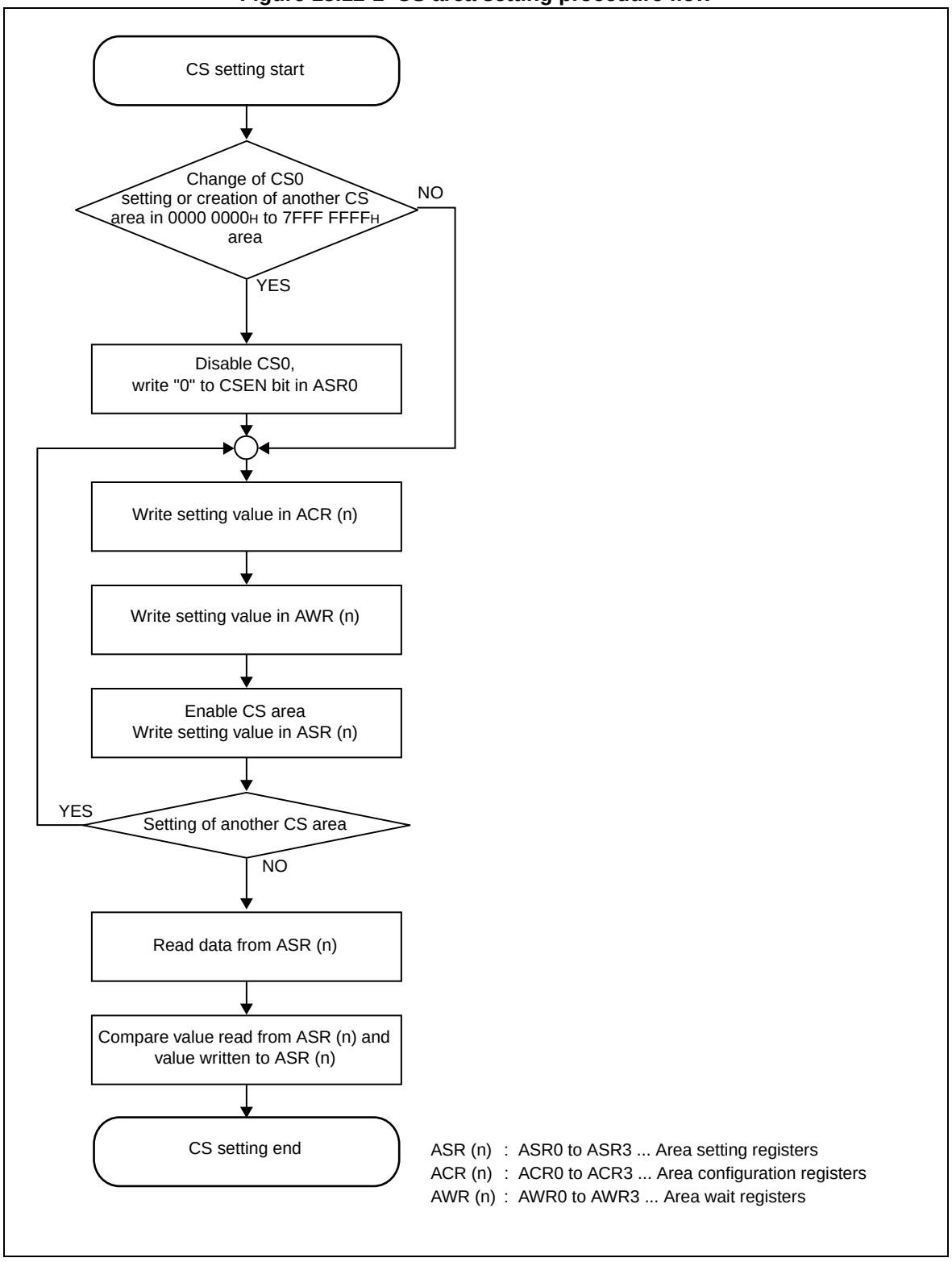
<Note>

Do not change the setting of a CS area while the CS area is being accessed.

■ Setting procedure

Figure 13.12-1 is a flowchart for a CS area setting procedure example.

Figure 13.12-1 CS area setting procedure flow


CS setting start
Change of CS0 setting or creation of another CS area in 0000 0000H to 7FFF FFFFH area
NO
YES
Disable CS0, write "0" to CSEN bit in ASR0
Write setting value in ACR (n)
Write setting value in AWR (n)
Enable CS area Write setting value in ASR (n)
YES
Setting of another CS area
NO
Read data from ASR (n)
Compare value read from ASR (n) and value written to ASR (n)
CS setting end
ASR (n) : ASR0 to ASR3 ... Area setting registers
ACR (n) : ACR0 to ACR3 ... Area configuration registers
AWR (n) : AWR0 to AWR3 ... Area wait registers

1. Write "0000 0000_H" to the area setting register (ASR0) through word access (only when changing the CS0 area or allocating another CS area to the 0000 0000_H to 7FFF $FFFF_H$ area).
2. Write a setting value in an area configuration register (ACR0 to ACR3) through word access.

   Use the area configuration register (ACR0 to ACR3) to make the following settings:
   - Data bus width
   - Address type
   - Bus type
3. Write a setting value to an area wait register (AWR0 to AWR3) through word access.
4. Write a setting value to area setting register (ASR0 to ASR3) through word access.

   Make the following settings with an area setting register (ASR0 to ASR3):
   - CS area
   - Write enable
   - Byte ordering (except for the CS0 area)
   - CS area enable/disable
5. To make settings for another CS area, repeat steps 2 to 4.
6. Read the area setting register (ASR0 to ASR3).
7. Compare the read values and the values that have been set in the area setting register (ASR0 to ASR3).

   Verify that the CS area settings are reflected in the subsequent access operations by reading the area setting register (ASR0 to ASR3) that was the last one set , and verify that the setting values and read values are the same.

<Notes>

- To change the CS0 area or allocate another CS area to the 0000 0000_H to 7FFF $FFFF_H$ area, first disable the CS0 area by using the CSEN bit (CSEN = 0) of the CS0 area setting register (ASR0).
- Notes on area setting register (ASR0 to ASR3) settings
  - Be sure that CS areas do not overlap one another. If any CS areas overlap, operation is not guaranteed.
  - The upper bits of the start address are set in the SADR31 to SADR16 bits. However, depending on the size of the area, the boundary is fixed in advance. The ASZ3 to ASZ0 bits that are valid bits must be set according to the CS area size. The SADR31 to SADR16 bits that are invalid must be set to "0".

■ CS area setting example

An example of setting values for the ASZ3 to ASZ0 bits and SADR31 to SADR16 bits in an area setting register (ASR0 to ASR3) and CS areas actually allocated is shown below.

- CS0 area settings

CS0 area setting register (ASR0): ASZ3 to ASZ0 = 0010_B

CS0 area setting register (ASR0): SADR31 to SADR16 = $000C_H$

→ $000C\ 0000_H$ to $000F\ FFFF_H$ is the CS0 area.

- CS1 area settings

CS1 area setting register (ASR1): ASZ3 to ASZ0 = 0000_B

CS1 area setting register (ASR1): SADR31 to SADR16 = 0006_H

→ $0006\ 0000_H$ to $0006\ FFFF_H$ is the CS1 area.

- CS2 area settings

A 1-MB space from $0011\ 0000_H$ is allocated.

The setting is ASZ3 to ASZ0 bits in the CS2 area setting register (ASR2) = 0100_B since a space of 1 MB must be prepared.

At this time, the SADR31 to SADR20 bits become valid and the SADR19 to SADR16 bits are excluded from the address comparison targets. Therefore, $0011\ 0000_H$ cannot be set as the start address of the CS2 area, and $0010\ 0000_H$ is set instead.

CS2 area setting register (ASR2): ASZ3 to ASZ0 = 0100_B

CS2 area setting register (ASR2): SADR31 to SADR16 = 0010_H

→ $0010\ 0000_H$ to $001F\ FFFF_H$ is the CS2 area.

- CS3 area settings

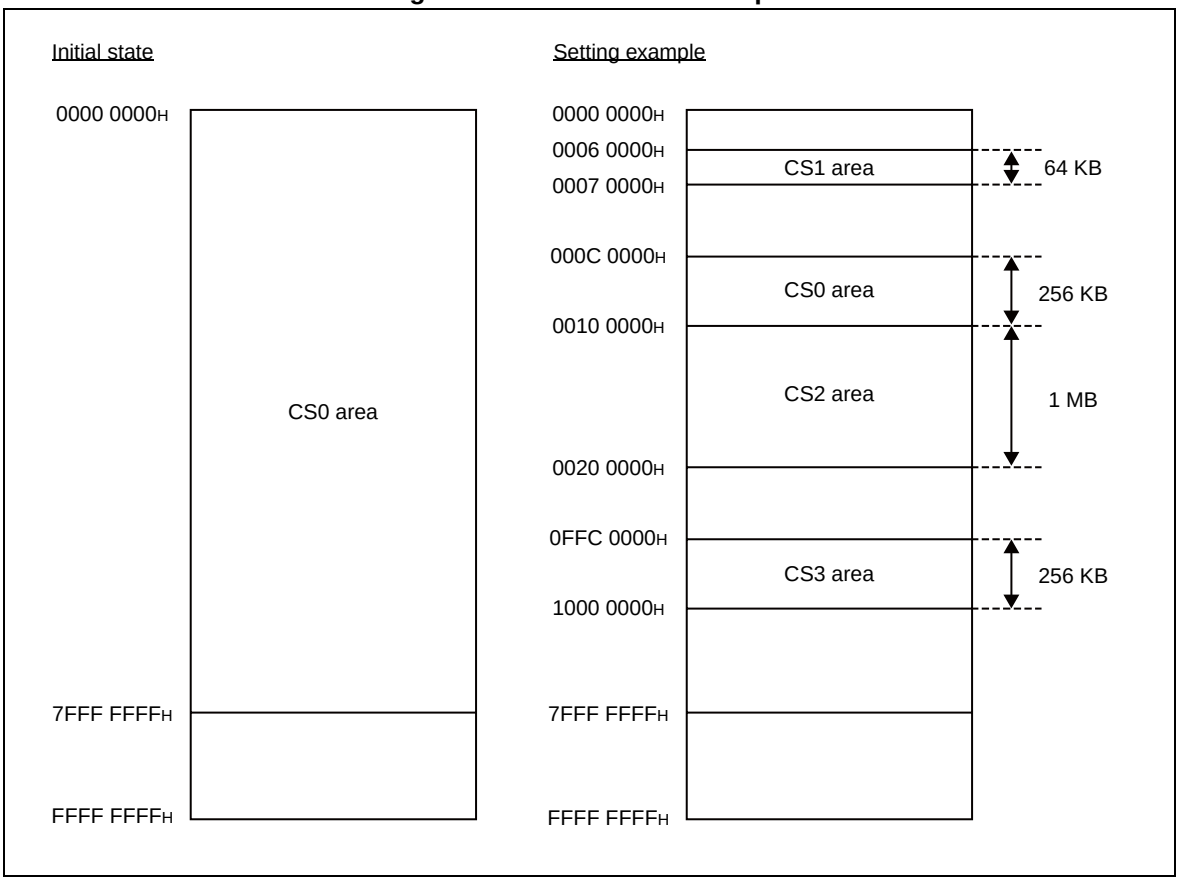
CS3 area setting register (ASR3): ASZ3 to ASZ0 = 0010_B

CS3 area setting register (ASR3): SADR31 to SADR16 = $0FFC_H$

→ $0FFC\ 0000_H$ to $0FFF\ FFFF_H$ is the CS3 area.

Figure 13.12-2 shows the CS areas in the above example.

Figure 13.12-2 CS area example



CHAPTER 14 Slave Interface

This chapter explains the functions and operations of the slave interface.

14.1 Overview
14.2 Configuration
14.3 Pins
14.4 Registers
14.5 Interrupts
14.6 Explanation of Operations and Setting Procedure Examples
14.7 Notes on Use

14.1 Overview

The slave interface is a port that serves as a slave bus when this series is used as a peripheral function (slave device) of an external CPU.
It consists of high-capacity transmission/reception FIFOs for transmitting/receiving large volumes of data and registers for controlling communication so that it can control communication even during data transmission/reception.
This can decrease the CPU load as well as make communication efficient.

■ Overview

The slave interface serves as a slave bus connected to an external CPU bus for data communication with an external CPU.

- Data communication: Communication can have a bit width of 16 bits.
- Built-in FIFO function: The slave interface has the following two types of built-in FIFO:
  - Reception FIFO

    The reception FIFO has built-in, dual-sided (A side and B side), single-port RAM. The number of FIFO stages that can be used is 768 (1536 bytes).

    One side can be written while the other side is being read.
  - Transmission FIFO

    The transmission FIFO has built-in, dual-port RAM. The number of FIFO stages that can be used 768 (1536 bytes).
- Transmission/reception registers: In addition to the FIFOs, 16-bit registers are provided separately for transmission and reception. Aside from data communication, they are used to control communication, such as receiving commands from external CPU or transmitting the status to external CPU.
- Interrupt request function: Interrupt requests can be generated in any of the following cases:
  - Data is written to the received data register that has been prepared for controlling communication.
  - Either the A side or B side of the reception FIFO is full.
  - The data in the transmission FIFO is empty.
- DMA transmission/reception function: The DMA controller (DMAC) can be used for DMA transfer of transmitted data and received data.

14.2 Configuration

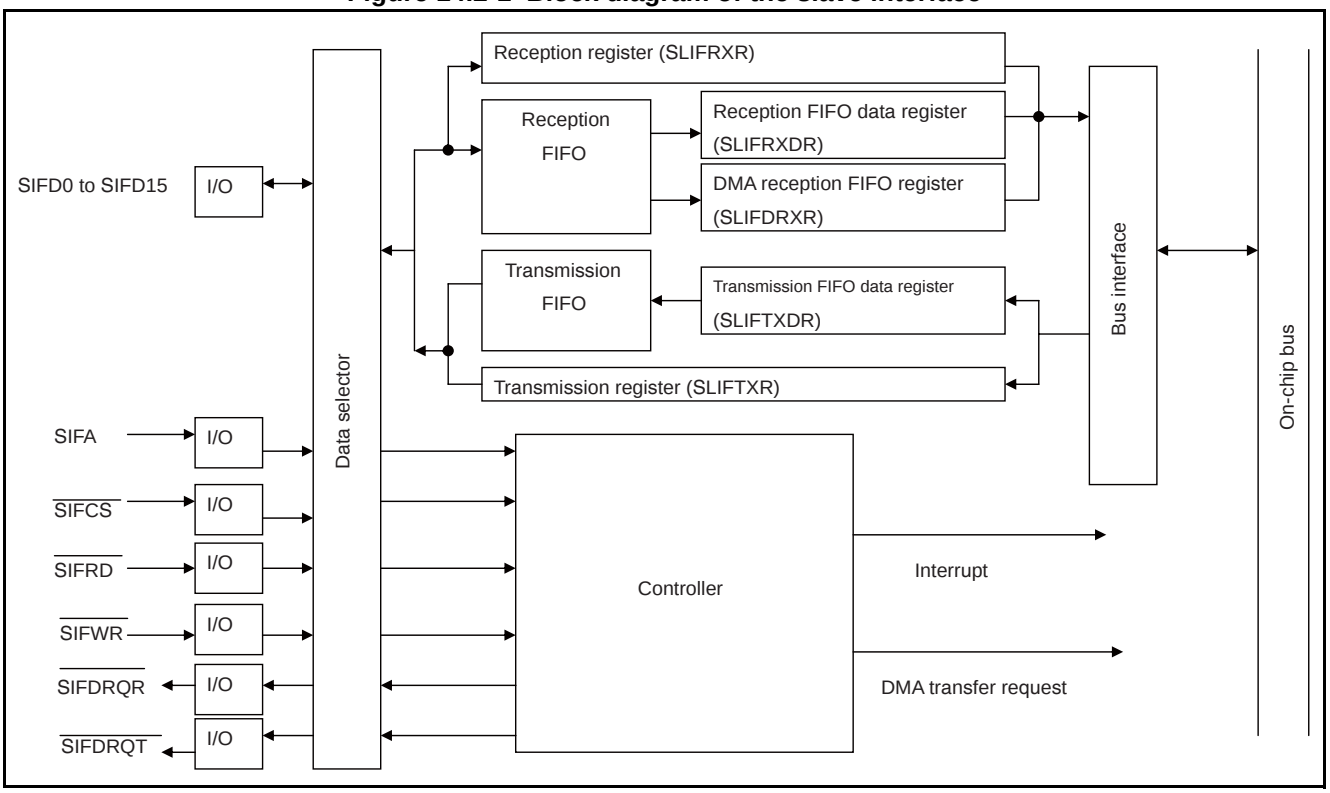
The slave interface consists of the following blocks:

- Reception FIFO
- Transmission FIFO
- Reception register (SLIFRXR)
- Transmission register (SLIFTXR)
- Bus interface
- Controller

■ Block diagram of the slave interface

Figure 14.2-1 is a block diagram of the slave interface.

Figure 14.2-1 Block diagram of the slave interface



- Reception FIFO

  This FIFO is a dual-sided reception FIFO.
- Transmission FIFO

  This FIFO is the transmission FIFO.
- Reception register (SLIFRXR)

  This received data register has been prepared for controlling communication in addition to the reception FIFO.
- Transmission register (SLIFTXR)

  This transmitted data register has been prepared for controlling communication in addition to the transmission FIFO.

- Controller

  The controller controls operations of the slave interface.

■ Clock

Table 14.2-1 lists the clock to be used for the slave interface.

Table 14.2-1 Clock used for the slave interface

Clock Name	Description
Operation clock	On-chip bus clock (HCLK)

14.3 Pins

This section explains the pins of the slave interface.

■ Overview

The slave interface has the following seven types of pin:

- SIFD0 to SIFD15 pins

  Data I/O pins for data communication with external CPUs. These pins are used as data bus bit0 to bit15.

  These pins are multiplexed pins. For details of using the SIFD0 to SIFD15 pins of the slave interface, see "2.4 Setting Method for Pins".

- $\overline{\text{SIFCS}}$ pin

  Chip select signal input pin

  This pin is a multiplexed pin. For details of using the $\overline{\text{SIFCS}}$ pin of the slave interface, see "2.4 Setting Method for Pins".

- SIFA pin

  Address signal input pin

  A signal input from this pin at the "L" level indicates communication control in communication with a transmission/reception register. A signal input from this pin at the "H" level indicates data communication in communication with the transmission/reception FIFO.

  This pin is a multiplexed pin. For details of using the SIFA pin of the slave interface, see "2.4 Setting Method for Pins".

- $\overline{\text{SIFRD}}$ pin

  Read strobe signal input pin

  This pin is a multiplexed pin. For details of using the $\overline{\text{SIFRD}}$ pin of the slave interface, see "2.4 Setting Method for Pins".

- $\overline{\text{SIFWR}}$ pin

  Write strobe signal input pin

  This pin is a multiplexed pin. For details of using the $\overline{\text{SIFWR}}$ pin of the slave interface, see "2.4 Setting Method for Pins".

- $\overline{\text{SIFDRQR}}$ pin

  Received data request signal output pin

  A received data request signal notifies an external CPU of the write-enabled state of the reception FIFO.

  This pin is a multiplexed pin. For details of using the $\overline{\text{SIFDRQR}}$ pin of the slave interface, see "2.4 Setting Method for Pins".

- $\overline{\text{SIFDRQT}}$ pin

  Transmitted data request signal output pin

  A transmitted data request signal notifies an external CPU that there is data in the transmission FIFO or transmission register (SLIFTXR).

  This pin is a multiplexed pin. For details of using the $\overline{\text{SIFDRQT}}$ pin of the slave interface, see "2.4 Setting Method for Pins".

14.4 Registers

This section explains the configuration and functions of registers of the slave interface.

■ Slave interface register list

Table 14.4-1 lists the registers of the slave interface.

Table 14.4-1 Slave interface registers

Abbreviated Register Name	Register Name	Reference
SLIFCR	Control register	14.4.1
SLIFSR	Status register	14.4.2
SLIFRXSR	Reception FIFO status register	14.4.3
SLIFTXSR	Transmission FIFO status register	14.4.4
SLIFRXDR	Reception FIFO data register	14.4.5
SLIFTXDR	Transmission FIFO data register	14.4.6
SLIFRXR	Reception register	14.4.7
SLIFTXR	Transmission register	14.4.8
SLIFDRXR	DMA reception FIFO register	14.4.9

14.4.1 Control Register (SLIFCR)

This register controls the reception register/FIFO and transmission register/FIFO.

Figure 14.4-1 shows the bit configuration of the control register (SLIFCR).

Figure 14.4-1 Bit configuration of the control register (SLIFCR)

bit	31							16
	Undefined							
Attribute	-							
Initial value	0							

bit	15						9	8
	Undefined							TX_DMA_ENA
Attribute	-							W
Initial value	0							0

bit	7	6	5	4	3	2	1	0
	RX_DMA_ENA	INT_TX_EMP	INT_RX_FULL	INT_RX_REG_I	TX_CLR	RX_CLR	TX_STAT	RX_STAT
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

W: Write only

<Note>

Be sure to write to this register in units of words.

[bit31 to bit9]: Undefined bits

In case of writing	Always write "0" to this (these) bit (bits).

[bit8]: TX_DMA_ENA (TX FIFO DMA transfer control bit)

This bit specifies whether a DMA transfer request signal is output to the DMA controller (DMAC) when no data remains in the transmission FIFO.

Written Value	Explanation
0	Does not output the request.
1	Outputs the request.

If this bit is set to "1" when no data remains in the transmission FIFO, data can be written to the transmission FIFO by DMA transfer.

<Note>

When the transmission FIFO is full, the DMA transfer request is cancelled.

[bit7]: RX_DMA_ENA (RX FIFO DMA transfer control bit)

This bit specifies whether a DMA transfer request signal is output to the DMA controller (DMAC) when either the A or B side of the reception FIFO is full.

The reception FIFO consists of two sides (A side and B side). A DMA transfer request signal is output each time that either side is full.

Written Value	Explanation
0	Does not output the request.
1	Outputs the request.

If this bit is set to "1" when either side of the reception FIFO is full, data in the reception FIFO can be transferred by DMA transfer.

<Note>

When the reception FIFO is empty, the DMA transfer request is cancelled.

[bit6]: INT_TX_EMP (TX FIFO empty interrupt control bit)

This bit specifies whether to generate a TX FIFO empty interrupt request when the transmission FIFO contains no data.

Written Value	Explanation
0	Disables generation of TX FIFO empty interrupt requests.
1	Enables generation of TX FIFO empty interrupt requests.

[bit5]: INT_RX_FULL (RX FIFO full interrupt control bit)

This bit specifies whether to generate an RX FIFO full interrupt request when either the A or B side of the reception FIFO is full.

The reception FIFO consists of two sides (A side and B side). A RX FIFO full interrupt request is generated each time that either side is full.

Written Value	Explanation
0	Disables generation of RX FIFO full interrupt requests.
1	Enables generation of RX FIFO full interrupt requests.

[bit4]: INT_RX_REG_I (RX register data input interrupt control bit)

This bit specifies whether to generate an RX register data input interrupt request when external data is written to the reception register (SLIFRXR).

Written Value	Explanation
0	Disables generation of RX register data input interrupt requests.
1	Enables generation of RX register data input interrupt requests.

[bit3]: TX_CLR (TX_REG/FIFO clear control bit)

This bit clears the transmission register (SLIFTXR) and transmission FIFO.

TX_CLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is always read.
1	Clears the value.	

<Note>

All the data in the transmission FIFO is cleared when "1" is written to this bit.

[bit2]: RX_CLR (RX_REG/FIFO clear control bit)

This bit clears the reception register (SLIFRXR) and reception FIFO.

RX_CLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is always read.
1	Clears the value.	

<Note>

All the data in the reception FIFO is cleared when "1" is written to this bit.

[bit1]: TX_STAT (External data transmission control bit)

This bit outputs a transmitted data request signal from the $\overline{\text{SIFDRQT}}$ pin and notifies an external CPU that there is transmitted data.

Written Value	Explanation
0	Does not output the transmitted data request signal.
1	Outputs the transmitted data request signal.

[bit0]: RX_STAT (External data reception control bit)

This bit outputs a received data request signal from the $\overline{\text{SIFDRQR}}$ pin to request the writing of data to an external CPU.

Written Value	Explanation
0	Does not output the received data request signal.
1	Outputs the received data request signal.

<Note>

If "0" is written to this bit to stop output of a received data request signal, clear the received data by writing "1" to the RX_CLR bit before writing "1" in this bit to output the received data request signal. Since all the data in the reception FIFO will be cleared at this time, read the data in the reception FIFO before writing "1" to the RX_CLR bit.

14.4.2 Status Register (SLIFSR)

This register indicates the reception register/FIFO status and the transmission register/FIFO status.

Figure 14.4-2 shows the bit configuration of the status register (SLIFSR).

Figure 14.4-2 Bit configuration of the status register (SLIFSR)

bit	31							8
	Undefined							
Attribute	-							
Initial value	0							

bit	7	6	5	4	3	2	1	0
	TX_FIFO1	TX_FIFO0	RX_FIFOB1	RX_FIFOB0	RX_FIFOA1	RX_FIFOA0	TX_REG_OUT	RX_REG_IN
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

<Note>

Be sure to read this register in units of words.

[bit31 to bit8]: Undefined bits

In case of reading	"0" is read.

[bit7, bit6]: TX_FIFO1, TX_FIFO0 (TX FIFO status bits)

These bits indicate whether the transmission FIFO contains data.

TX_FIFO1	TX_FIFO0	Explanation
0	0	The transmission FIFO contains no data. Alternatively, the transmission FIFO has been cleared.
0	1	The transmission FIFO contains data.
1	0	The transmission FIFO is full.
1	1	Reserved

If the INT_TX_EMP bit in the control register (SLIFCR) is set to "1" and these bits are "00", a TX FIFO empty interrupt request is generated.

<Note>

These bits become "00" when "1" is written to the TX_CLR bit in the control register (SLIFCR) and the transmission FIFO is cleared.

[bit5, bit4]: RX_FIFOB1, RX_FIFOB0 (RX FIFO B side status bits)

These bits indicate whether the B side of the reception FIFO contains data.

RX_FIFOB1	RX_FIFOB0	Explanation
0	0	The B side of the reception FIFO contains no data. Alternatively, the B side of the reception FIFO has been cleared.
0	1	The B side of the reception FIFO contains data.
1	0	The B side of the reception FIFO is full.
1	1	Reserved

If the INT_RX_FULL bit in the control register (SLIFCR) is set to "1" and these bits are "10", an RX FIFO full interrupt request is generated.

<Note>

These bits become "00" when "1" is written to the RX_CLR bit in the control register (SLIFCR) and the reception FIFO is cleared.

[bit3, bit2]: RX_FIFOA1, RX_FIFOA0 (RX FIFO A side status bits)

These bits indicate whether the A side of the reception FIFO contains data.

RX_FIFOA1	RX_FIFOA0	Explanation
0	0	The A side of the reception FIFO contains no data. Alternatively, the A side of the reception FIFO has been cleared.
0	1	The A side of the reception FIFO contains data.
1	0	The A side of the reception FIFO is full.
1	1	Reserved

If the INT_RX_FULL bit in the control register (SLIFCR) is set to "1" and these bits are "10", an RX FIFO full interrupt request is generated.

<Note>

These bits become "00" when "1" is written to the RX_CLR bit in the control register (SLIFCR) and the reception FIFO is cleared.

[bit1]: TX_REG_OUT (SLIF TX register output status bit)

This bit indicates whether data has been output from the transmission register (SLIFTXR).

Read Value	Explanation
0	Data has not been output. Alternatively, the transmission register (SLIFTXR) has been cleared.
1	Data has been output.

<Note>

This bit becomes "0" when the CPU has completely read this register or when "1" is written to the TX_CLR bit in the control register (SLIFCR) to clear the transmission register (SLIFTXR).

[bit0]: RX_REG_IN (SLIF RX register input status bit)

This bit indicates whether the reception register (SLIFRXR) has received external data.

Read Value	Explanation
0	The data has not been received. Alternatively, the reception register (SLIFRXR) has been cleared.
1	The data has been received.

If the INT_RX_REG_I bit in the control register (SLIFCR) is set to "1" and this bit is "1", an RX register data input interrupt request is generated.

<Note>

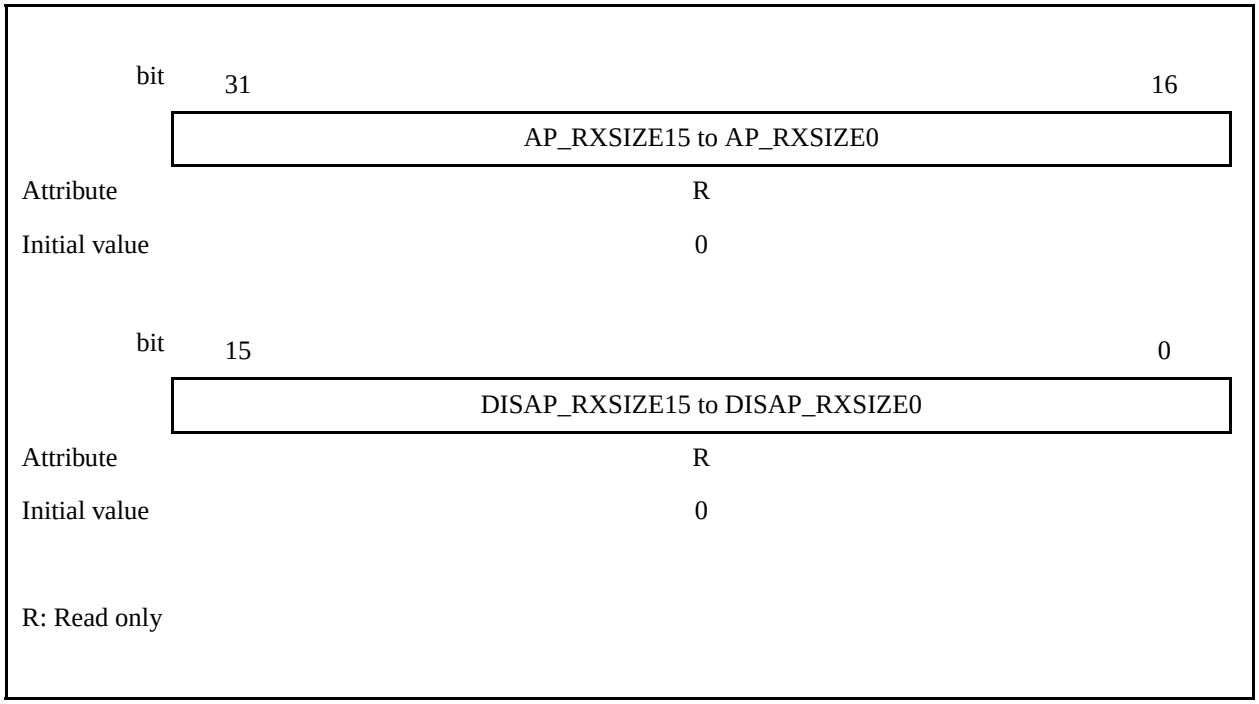
This bit becomes "0" when the CPU has completely read this register or when "1" is written to the RX_CLR bit in the control register (SLIFCR) to clear the reception register (SLIFRXR).

14.4.3 Reception FIFO Status Register (SLIFRXSR)

This register indicates the reception FIFO status.

Figure 14.4-3 shows the bit configuration of the reception FIFO status register (SLIFRXSR).

Figure 14.4-3 Bit configuration of the reception FIFO status register



<Note>

Be sure to read this register in units of words.

[bit31 to bit16]: AP_RXSIZE15 to AP_RXSIZE0 (RX FIFO read enabled data size bits)

The reception FIFO is a dual-sided FIFO where data can be read from one side while data is written to the other side, and it is also referred to as a ping-pong FIFO.

These bits specify the size of data, in units of bytes, that can be read when either the A or B side or both sides of the reception FIFO are full.

<Note>

If more data than specified by the bits is read, the correct data cannot be read.

[bit15 to bit0]: DISAP_RXSIZE15 to DISAP_RXSIZE0 (RX FIFO read disabled data size bits)

These bits indicate the size of data written to the reception FIFO from an external CPU.

To prevent a conflict between the read operation and the write operation, these bits indicate the size of data, in units of bytes, that is written on the side connected to the external CPU until that side is full.

<Note>

For details of reading data when a read-disabled data size is indicated by these bits, such as when reading the data written in the reception FIFO that is not yet full, see "14.6.1 Explanation of Operations of the Slave Interface".

14.4.4 Transmission FIFO Status Register (SLIFTXSR)

This register indicates the size of data that can be written to the transmission FIFO.
Since the transmission FIFO consists of dual-port RAM, data can be read from an external CPU while data is written to the transmission FIFO.

Figure 14.4-4 shows the bit configuration of the transmission FIFO status register (SLIFTXSR).

Figure 14.4-4 Bit configuration of the transmission FIFO status register

bit	31															16
	AP_TXSIZE15 to AP_TXSIZE0															
Attribute	R															
Initial value	0															

bit	15															0
	Undefined															
Attribute	0															
Initial value	-															

R: Read only

<Note>

Be sure to read this register in units of words.

[bit31 to bit16]: AP_TXSIZE15 to AP_TXSIZE0 (TX FIFO write enabled data size bits)

These bits indicate the size of data, in units of bytes, that can be written to the transmission FIFO.

[bit15 to bit0]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

14.4.5 Reception FIFO Data Register (SLIFRXDR)

This register reads data from the reception FIFO.
The register can read data from the reception FIFO.

Figure 14.4-5 shows the bit configuration of the reception FIFO data register (SLIFRXDR).

Figure 14.4-5 Bit configuration of the reception FIFO data register (SLIFRXDR)

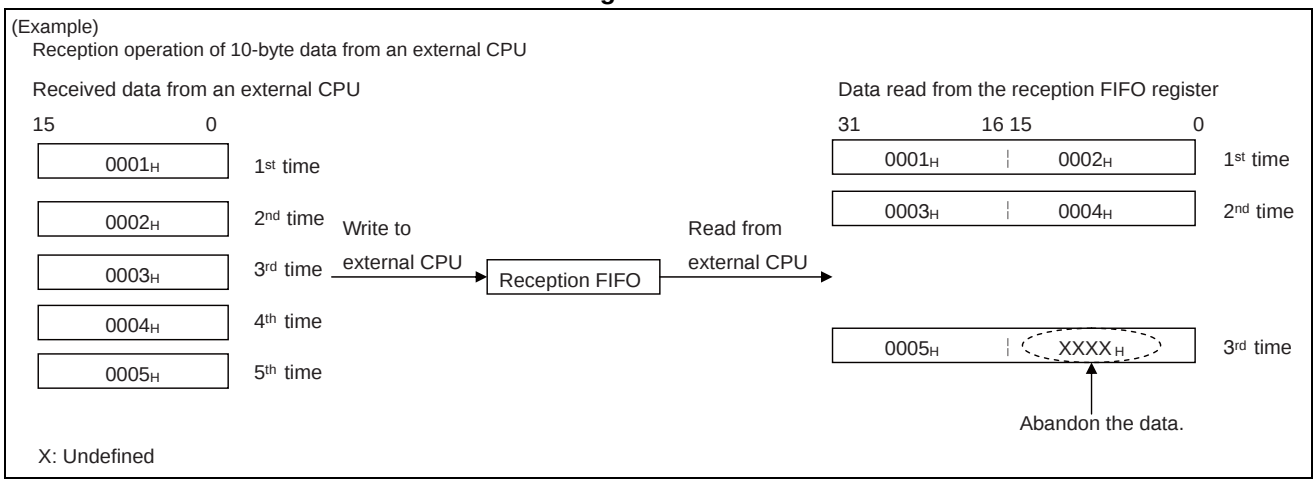
bit	31															0
	RD31 to RD0															
Attribute	R															
Initial value	0															

R: Read only

<Notes>

- The data (16-bit width) written to the reception FIFO from an external CPU is read in order of upper bits [31:16] to lower bits [15:0] of this register in the reception FIFO.
- In case of reading the data written to the reception FIFO from an external CPU for odd number of times, the lower bits [15:0] of the final data is undefined; therefore, the data should be abandoned.
- Be sure to read this register in units of words.
- The reception data cannot be read from this register by DMA transfer. Read that data from the DMA reception FIFO register (SLIFDRXR).

Figure 14.4-6



14.4.6 Transmission FIFO Data Register (SLIFTXDR)

This register writes data to the transmission FIFO.
The value written to the register will be written to the transmission FIFO.

Figure 14.4-7 shows the bit configuration of the transmission FIFO data register (SLIFTXDR).

Figure 14.4-7 Bit configuration of the transmission FIFO data register (SLIFTXDR)

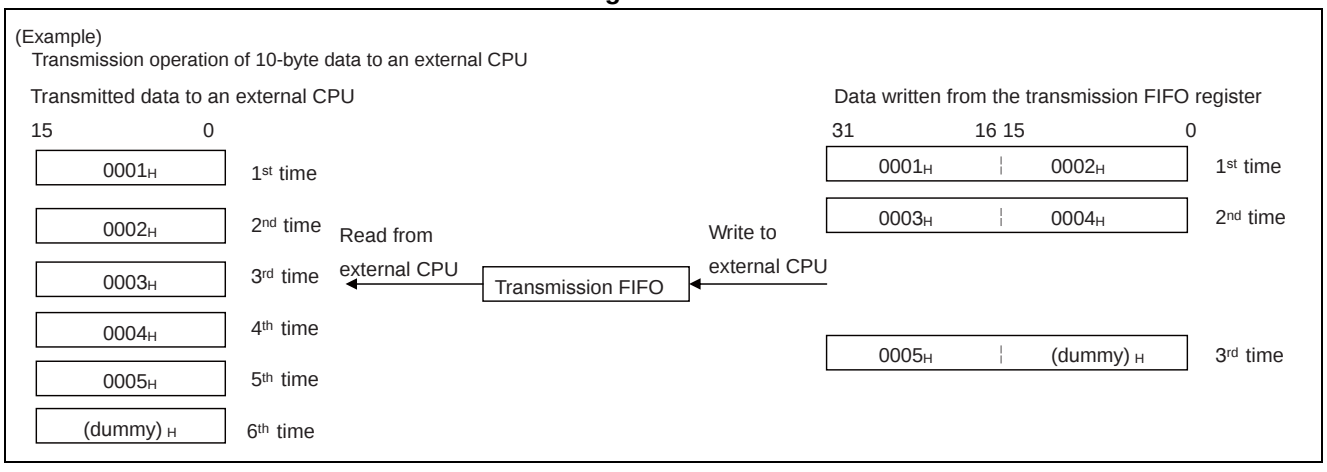
bit	31	0
	TD31 to TD0	
Attribute	W	
Initial value	0	

W: Write only

<Notes>

- The data (16-bit width) read from the transmission FIFO from an external CPU is written in order of upper bits [31:16] to lower bits [15:0] of this register in the transmission FIFO.
- In case of an external CPU reads the data from the transmission FIFO, the data should be read for even number of times.
- Be sure to write to this register in units of words.

Figure 14.4-8

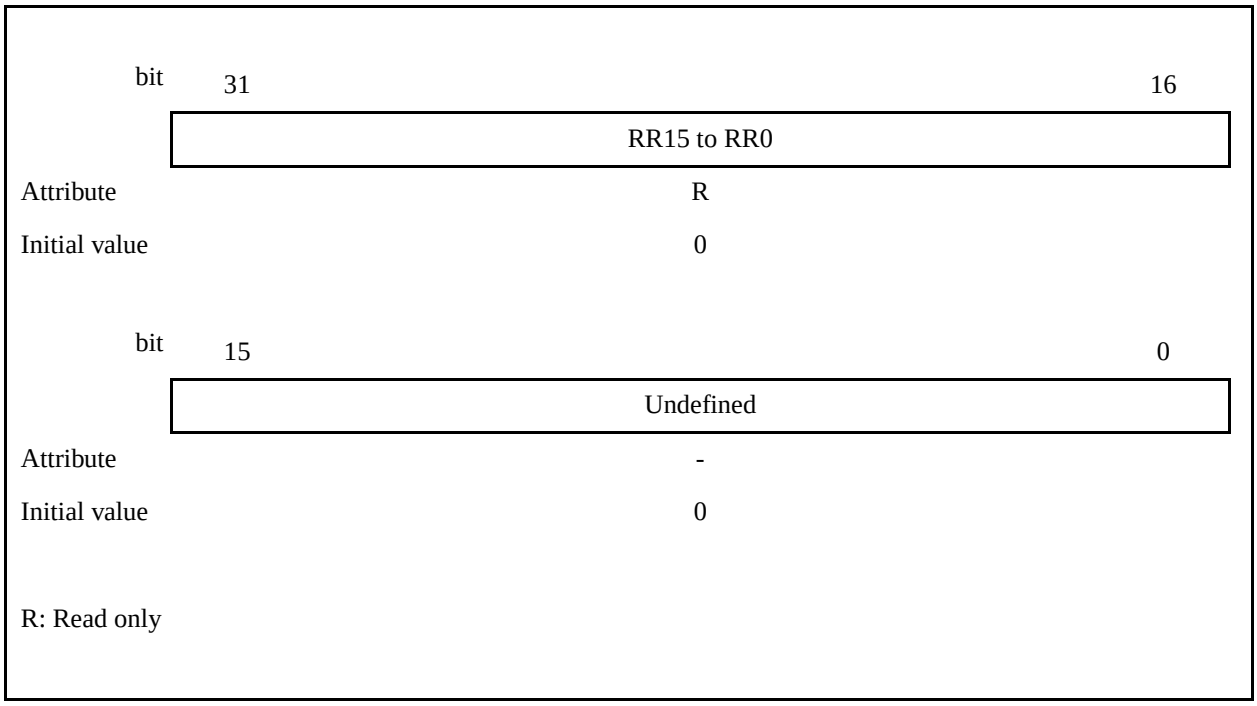


14.4.7 Reception Register (SLIFRXR)

This register reads data that has been written from an external CPU. The register is provided separately from the reception FIFO to control communication.

Figure 14.4-9 shows the bit configuration of the reception register (SLIFRXR).

Figure 14.4-9 Bit configuration of the reception register (SLIFRXR)



[bit31 to bit16]: RR15 to RR0 (reception register data bit)

These bits can read the written data from an external CPU.

[bit15 to bit0]: Undefined bits

In case of reading	"0" is read.

<Note>

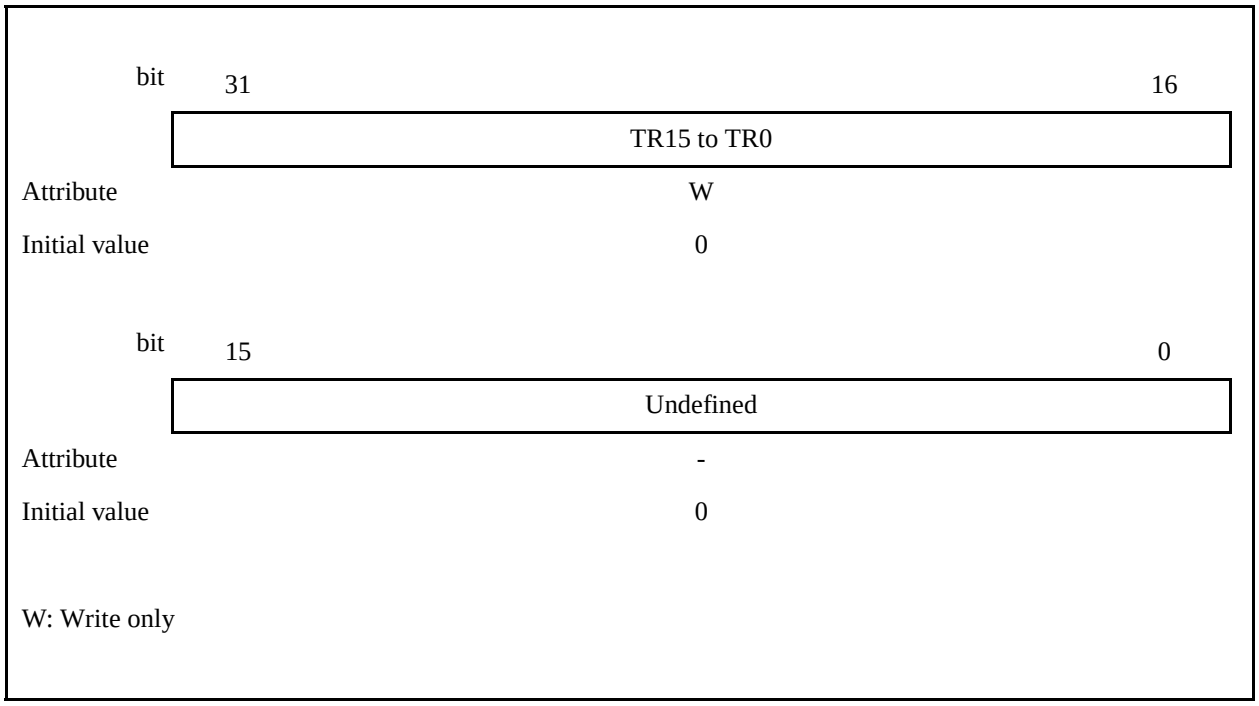
Be sure to read this register in units of words.

14.4.8 Transmission Register (SLIFTXR)

This register writes data that has been read by an external CPU. The register is provided separately from the transmission FIFO to control communication.

Figure 14.4-10 shows the bit configuration of the transmission register (SLIFTXR).

Figure 14.4-10 Bit configuration of the transmission register (SLIFTXR)



[bit31 to bit16]: TR15 to TR0 (transmission register data bit)

These bits can write the data read by an external CPU.

[bit15 to bit0]: Undefined bits

In case of writing	Ignored

<Note>

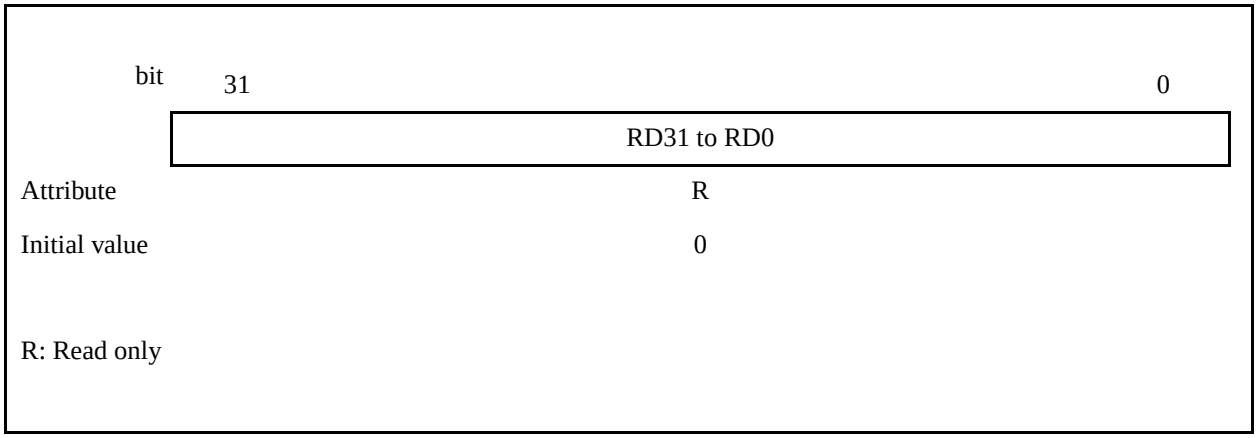
Be sure to write to this register in units of words.

14.4.9 DMA Reception FIFO Register (SLIFDRXR)

This register reads data in the reception FIFO by DMA transfer.

Figure 14.4-11 shows the bit configuration of the DMA reception FIFO register (SLIFDRXR).

Figure 14.4-11 Bit configuration of the DMA reception FIFO register (SLIFDRXR)



<Notes>

- Be sure to read this register in units of words.
- This register is a dedicated DMA transfer for the reception FIFO read register. Do not read from this register by CPU.

14.5 Interrupts

An interrupt request is generated under the following conditions:

- Data has been written to the reception register (SLIFRXR) (reception register data input interrupt request).
- Either side of the reception FIFO is full (reception FIFO full interrupt request).
- The transmission FIFO contains no data (transfer FIFO empty interrupt request).

■ Reception interrupt request

Table 14.5-1 lists the conditions under which a reception interrupt request is generated.

Table 14.5-1 Reception interrupt request generation conditions

Interrupt Request		Interrupt source (SLIFSR)	Interrupt Request Enabled (SLIFCR)	Clearing of Interrupt Request
Reception register data input interrupt request		RX_REG_IN = 1	INT_RX_REG_I = 1	Data is read from the SLIFSR.
Reception FIFO full interrupt request	FIFO A	RX_FIFOA1, RX_FIFOA0 = 10	INT_RX_FULL = 1	After reading the data of the A side of the reception FIFO, data is read from the SLIFSR.
	FIFO B	RX_FIFOB1, RX_FIFOB0 = 10	INT_RX_FULL = 1	After reading the data of the B side of the reception FIFO, data is read from the SLIFSR.

SLIFSR: Status register

SLIFCR: Control register

<Notes>

- If generation of the interrupt requests is enabled when the interrupt source is established, an interrupt request is generated simultaneously.

  Perform the following operation for enabling generation of the interrupt requests:

  - Clear interrupt requests before enabling generation of the interrupt requests.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set an interrupt level corresponding to the interrupt vector number. For details of setting an interrupt level, see "CHAPTER 10 Interrupt Controller".

■ Transmission interrupt request

Table 14.5-2 lists the conditions under which a transmission interrupt request is generated.

Table 14.5-2 Transmission interrupt request generation conditions

Interrupt Request	Interrupt source (SLIFSR)	Interrupt Request Enabled (SLIFCR)	Clearing of Interrupt Request
Transmission FIFO empty interrupt request	TX_FIFO1, TX_FIFO0 = 00	INT_TX_EMP bit = 1	After writing data to the transmission FIFO, data is read from the SLIFSR.

SLIFSR: Status register

SLIFCR: Control register

<Notes>

- If generation of the interrupt requests is enabled when the interrupt source is established an interrupt request is generated simultaneously.

  Perform the following operation for enabling generation of the interrupt requests:

  - Clear interrupt requests before enabling generation of the interrupt requests.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set an interrupt level corresponding to the interrupt vector number. For details of setting an interrupt level, see "CHAPTER 10 Interrupt Controller".

■ Activating a DMA transfer

A DMA transfer can be activated when a slave interface DMA transfer request is generated.

For details of DMA transfer, see "CHAPTER 33 DMA Controller (DMAC)".

14.6 Explanation of Operations and Setting Procedure Examples

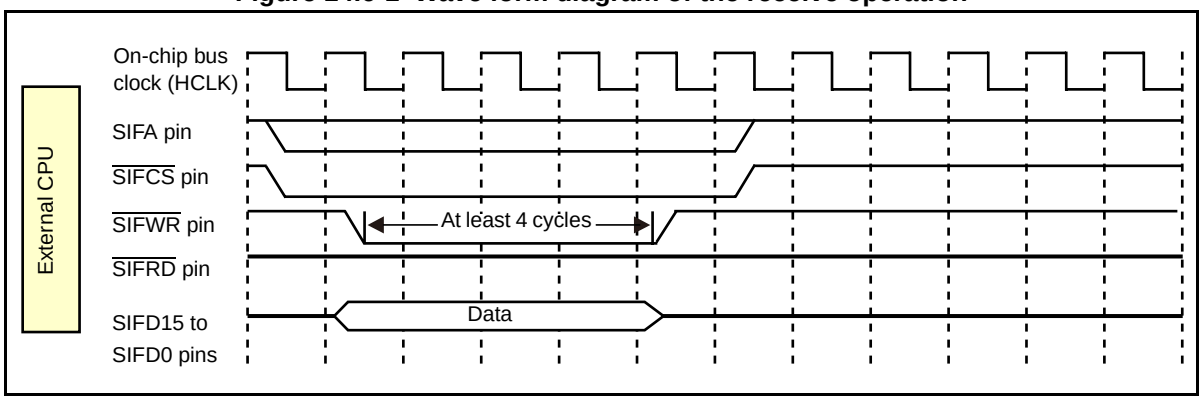
This section explains operations of the slave interface.

14.6.1 Explanation of Operations of the Slave Interface

■ Receive operation

Figure 14.6-1 is a wave form diagram of the receive operation.

Figure 14.6-1 Wave form diagram of the receive operation



● Reception register (SLIFRXR) operations

If the signal (address signal) input from the SIFA pin is at the "L" level when data is written from an external CPU, that data is stored in the reception register (SLIFRXR).

The operation of writing to the reception register (SLIFRXR) is explained below.

1. "1" is written to the RX_STAT bit in the control register (SLIFCR).

   A received data request signal is output from the $\overline{\text{SIFDRQR}}$ pin.
2. The external CPU writes data.

   A signal at the "L" level is input from the SIFA pin at this time.
3. The written data is stored in the reception register (SLIFRXR).
4. The RX_REG_IN bit in the status register (SLIFSR) changes to "1".

   At this time, if the INT_RX_REG_I bit in the control register (SLIFCR) is set to "1", an RX register data input interrupt request is generated.

The received data can be verified through reading from the reception register (SLIFRXR).

<Notes>

- Writing "0" to the RX_STAT bit in the control register (SLIFCR) cancels the received data request. Unless the received data request signal is output, the data written from the external CPU is not stored in the reception register (SLIFRXR).
- If data is written continuously to the reception register (SLIFRXR) from the external CPU, the next data may be written before the last written data in the reception register (SLIFRXR) is read.

  Some software processing to prevent this problem should be executed, such as reflecting the data read from the reception register (SLIFRXR) to the transmission register (SLIFTXR).

● Reception FIFO operations

If the signal (address signal) input from the SIFA pin is at the "H" level when data is written from an external CPU, that data is stored in the reception FIFO.

The operation of writing to the reception FIFO is explained below.

1. "1" is written to the RX_STAT bit in the control register (SLIFCR).

   A received data request signal is output from the $\overline{\text{SIFDRQR}}$ pin.

2. The external CPU writes data.

   A signal at the "H" level is input from the SIFA pin at this time.

3. The written data is stored in the reception FIFO.

   Data is alternately stored on the A side and B side of the reception FIFO.

4. The bits in the status register (SLIFSR) change as follows:

   If the A side of the reception FIFO is full: The RX_FIFOA1 and RX_FIFOA0 bits change to "10".

   If the B side of the reception FIFO is full: The RX_FIFOB1 and RX_FIFOB0 bits change to "10".

   At this time, if the INT_RX_FULL bit in the control register (SLIFCR) is set to "1", an RX FIFO full interrupt request is generated.

The received data can be verified through reading from the reception FIFO data register (SLIFRXDR).

<Notes>

- The reception FIFO that is not full can be verified to determine whether it has written data by reading the DISAP_RXSIZE15 to DISAP_RXSIZE0 bits in the reception FIFO status register (SLIFRXSR).
- When reading data written to the reception FIFO that is not full, stop receiving data by writing "0" to the RX_STAT bit in the control register (SLIFCR).

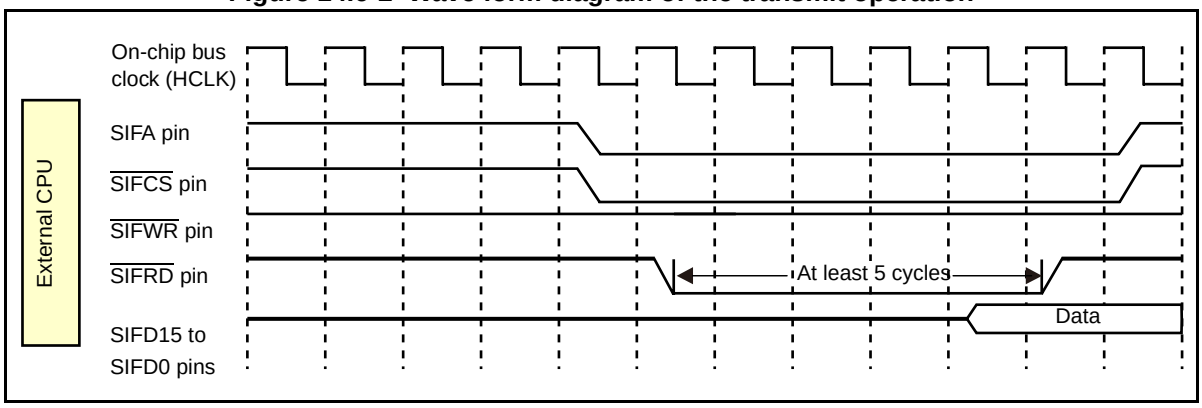
  However, when writing "1" back to the RX_STAT bit in the control register (SLIFCR) after stopping data reception, set the RX_CLR bit in the control register (SLIFCR) to clear the FIFO.

  Since clearing the reception FIFO will clear all the data in the FIFO, read the data in the reception FIFO before clearing it.
- The received data request signal is cancelled in the following cases:
  - Both the A and B sides of the reception FIFO are full.
  - "0" is written to the RX_STAT bit in the control register (SLIFCR).
- Unless the received data request signal is output, the data written from the external CPU is not stored in the reception FIFO.

■ Transmit operation

Figure 14.6-2 is a wave form diagram of the transmit operation.

Figure 14.6-2 Wave form diagram of the transmit operation



● Transmission register (SLIFTXR) operations

If the address signal (SIFA) is at the "L" level when an external CPU executes processing to read data from this device, it will read data from the transmission register (SLIFTXR).

The read operation on the transmission register (SLIFTXR) is explained below.

1. "1" is written to the TX_STAT bit in the control register (SLIFCR).

   A transmitted data request signal (SIFDRQT) is enabled.
2. The external CPU reads data from the transmission register (SLIFTXR).
3. The TX_REG_OUT bit in the status register (SLIFSR) changes to "1".

   Once the external CPU has completely read the transmission register (SLIFTXR), it cannot read data from the transmission register (SLIFTXR) until the next time that data is written to the transmission register (SLIFTXR).
4. The transmitted data request signal (SIFDRQT) is disabled when all the data in the transmission register (SLIFTXR) has been read.

<Notes>

- To stop transmission, write "1" to the TX_CLR bit in the control register (SLIFCR), and then write "0" to the TX_STAT bit.
- While the transmitted data request signal (SIFDRQT) is disabled, data cannot be read from the transmission register (SLIFTXR) even if an external read operation is performed.

● Transmission FIFO operations

If the address signal (SIFA) is at the "H" level when an external CPU executes processing to read data from this device, it will read data from the transmission FIFO.

The read operation on the transmission FIFO is explained below.

1. "1" is written to the TX_STAT bit in the control register (SLIFCR) to write data to the TX_FIFO (transmission FIFO) in this product.

   The transmitted data request signal (SIFDRQT) is enabled, and a data read request is issued to the external CPU.
2. The external CPU reads data from the transmission FIFO.
3. Once all the data in the TX_FIFO (transmission FIFO) has been read, the TX_FIFO1 and TX_FIFO0 bits in the status register (SLIFSR) change to "00", and the transmitted data request signal (SIFDRQT) is disabled.

   At this time, if the INT_TX_EMP bit in the control register (SLIFCR) is set to "1", an interrupt request is generated.

<Notes>

- To stop transmission, write "1" to the TX_CLR bit in the control register (SLIFCR), and then write "0" to the TX_STAT bit.
- While the transmitted data request signal (SIFDRQT) is disabled, data cannot be read from the transmission FIFO even if an external read operation is performed.

14.7 Notes on Use

Note the following points about using the slave interface.

■ Notes on operations

- If "0" is written to the RX_STAT bit in the control register (SLIFCR) to stop output of a received data request signal (SIFDRQR), clear the received data by writing "1" to the RX_CLR bit before writing "1" to the RX_STAT bit to output the received data request signal. Since all the data in the reception FIFO will be cleared at this time, read the data in the reception FIFO before writing "1" to the RX_CLR bit.
- If the size of the read data from the reception FIFO is greater than the read enabled data size, which is indicated by the reception FIFO status register (SLIFRXSR), the correct data cannot be read.
- While the received data request signal (SIFDRQR) is disabled, data is not written to the reception register (SLIFRXR) and reception FIFO even if a write operation is performed by an external CPU.
- While the transmitted data request signal (SIFDRQT) is disabled, data is not read from the transmission register (SLIFTXR) and transmission FIFO even if a read operation is performed by an external CPU.

CHAPTER 15 I/O Ports

This chapter explains the functions and operations of the I/O ports.

15.1 Overview
15.2 Configuration
15.3 Pins
15.4 Registers
15.5 Notes on Use

15.1 Overview

Pins of this series that are not used for the external bus interface or peripheral functions can be used as I/O ports.
This series is equipped with 99 I/O ports.

■ Overview

The I/O ports have the following features:

- Each pin can be specified as an I/O port used only as an input port or output port.
- Each pin can be specified as a pin used as an I/O port or a pin for a peripheral function or the external bus interface.

  Also, one of the I/O modes listed below can be selected depending on the register settings:

  Table 15.1-1 lists the I/O modes.

Table 15.1-1 I/O modes

I/O mode	Access to PDR	
Port input mode	In case of reading (except RMW instructions)	The levels of external pins are read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR.
Port output mode	In case of reading (except RMW instructions)	The PDR value is read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR and output to an external pin.
Peripheral function output mode *	In case of reading (except RMW instructions)	The output level from a peripheral function or the PDR value is read.
	In case of reading (RMW instructions)	The PDR value is read.
	In case of writing	The written value is stored in a PDR.

PDR: Port data register (PDR0 to PDRK)
RMW instruction: Read-modify-write instruction
* : The value that is read varies depending on the register settings.

- A pull-up resistor can be set for each pin.
- If Hi-Z is set to a pin with the CPU in standby mode (stop mode/watch mode), input is fixed at "0". However, input is not fixed at "0" for external interrupt requests whose generation is enabled and it can be used.

- A peripheral function can be assigned to any pin available for peripheral functions, if more than one pin is available, and peripheral function output from the pin can be enabled/disabled.

  However, if the peripheral function has more than one I/O, each I/O must be set to individual ports belonging to the same group.
- Example: Ch.1 multifunction serial interface settings

Serial Data Output	Serial Clock I/O	Serial Data Input	Valid Port
SOUT1 pin (Port 0)	SCK1 pin (Port 0)	SIN1 pin (Port 0)	Port 0
		SIN1_1 pin (Port 1)	Setting prohibited
	SCK1_1 pin (Port 1)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	
SOUT1_1 pin (Port 1)	SCK1 pin (Port 0)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	
	SCK1_1 pin (Port 1)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	Port 1

15.2 Configuration

This series has the following 3 types of built-in I/O port:
- Ordinary I/O ports
- Analog input multifunction I/O ports
- Analog output multifunction I/O ports

■ Overview

3 types of built-in I/O port that this series has are described below.

- Ordinary I/O ports

  These I/O ports have basic configurations in which the ports are used also for I/O of peripheral functions. Each port consists of the following blocks:

  - Port function registers (PFR0 to PFRH)
  - Port data direction registers (DDR0 to DDRK)
  - Extended port function registers (EPFR0 to EPFR35)
  - Pull-up resistor control registers (PCR0 to PCRK)
  - Port data registers (PDR0 to PDRK)

- Analog input multifunction I/O ports

  These I/O ports are used also for analog input of the 10-bit A/D converter. Each port consists of an analog input enable block and the ordinary I/O port blocks.

  The analog input multifunction ports are P77 to P70, P87 to P80, and PA7 to PA0.

- Analog output multifunction I/O ports

  These I/O ports are used also for analog output of the 8-bit D/A converter. Each port consists of an analog output enable block and the ordinary I/O port blocks, except those of the following registers:

  - Port function registers (PFR0 to PFRH)
  - Extended port function registers (EPFR0 to EPFR35)

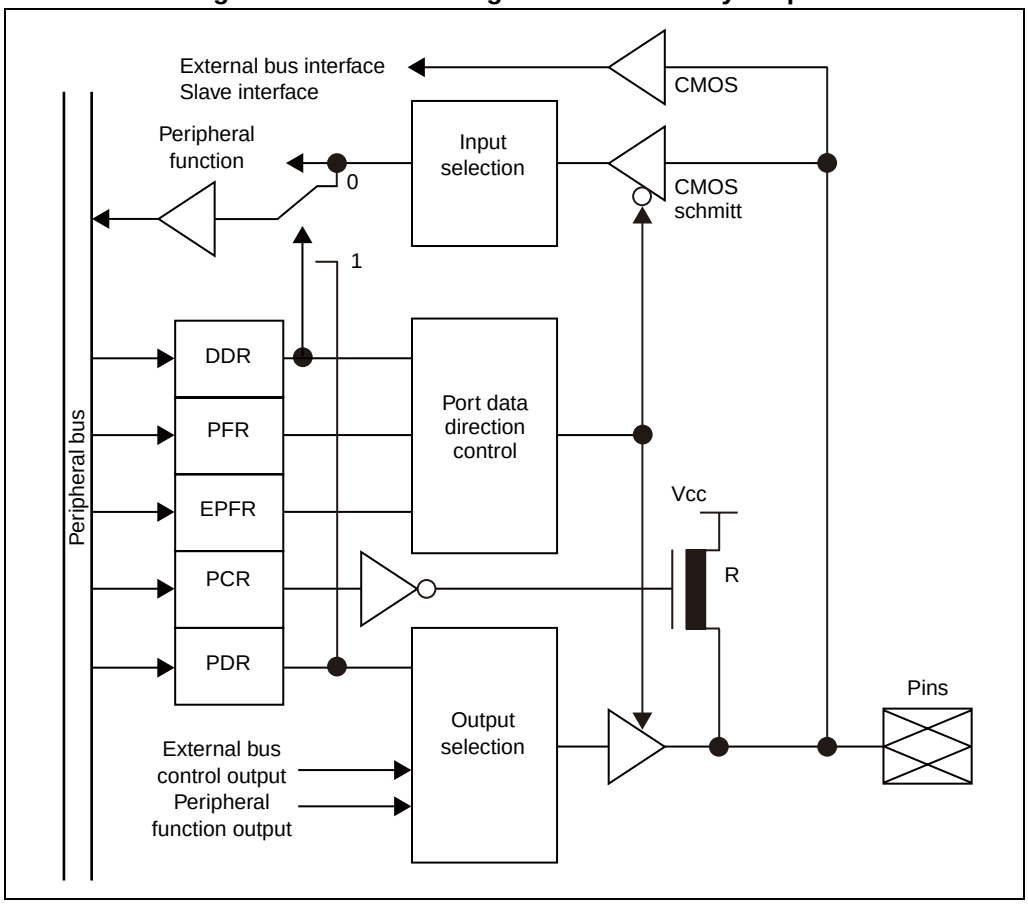
  The analog output multifunction ports are P92 to P90.

■ Block diagrams

● Ordinary I/O ports

Figure 15.2-1 is a block diagram of an ordinary I/O port.

Figure 15.2-1 Block diagram of an ordinary I/O port



- Port data direction registers (DDR0 to DDRK)

  These registers set the I/O directions of pins used as general-purpose ports.

  For a pin for a peripheral function or the external bus interface, these registers set the contents read from a port data register (PDR0 to PDRK).

- Port function registers (PFR0 to PFRH)

  These registers select how to use individual pins.

- Extended port function registers (EPFR0 to EPFR35)

  These registers set the pin to which a peripheral function is assigned from among the multiple pins available for peripheral functions. Peripheral function output from such pins is enabled/disabled according to the registers.

- Pull-up resistor control registers (PCR0 to PCRK)

  These registers set pull-up resistors. With one register provided for each port, a pull-up resistor can be connected to each pin.

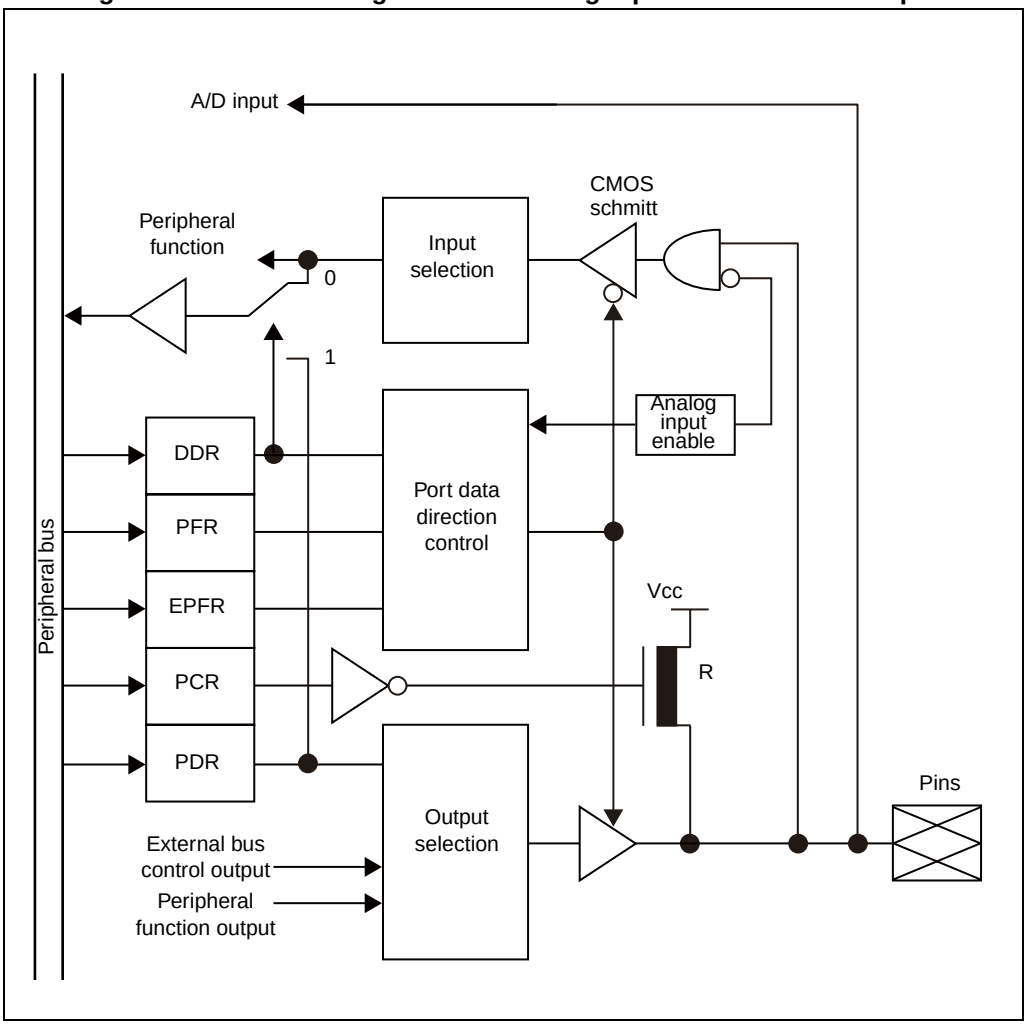
- Port data registers (PDR0 to PDRK)

  These registers store output data. The meanings of read and written values vary depending on the mode of the port.

● Analog input multifunction I/O port

Figure 15.2-2 is a block diagram of an analog input multifunction I/O port.

Figure 15.2-2 Block diagram of an analog input multifunction I/O port



The analog input multifunction I/O port consists of the blocks that are components of each ordinary I/O port and the analog input enable block.

This block enables analog input from pins for which input is enabled by the A/D channel enable register (ADCHE).

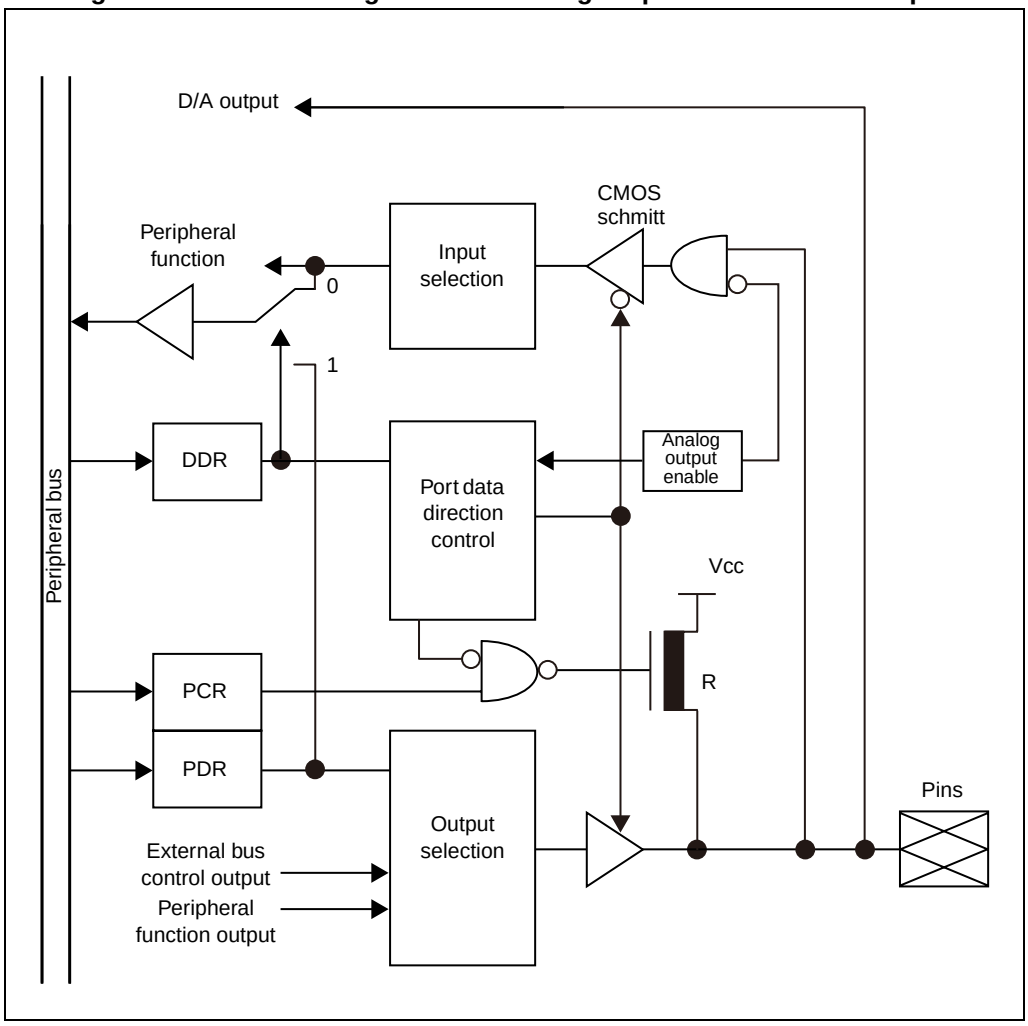
<Notes>

- The analog input multifunction ports are P77 to P70, P87 to P80, and PA7 to PA0.
- In serial write mode selected by the MD1 and MD0 pins (MD1, MD0 = 01), digital input is enabled and analog input is disabled only for P75 (AN5 pin).

● Analog output multifunction I/O ports

Figure 15.2-3 is a block diagram of an analog output multifunction I/O port.

Figure 15.2-3 Block diagram of an analog output multifunction I/O port



The analog output multifunction I/O port consists of the blocks that are components of each ordinary I/O port other than the following register and the analog output enable block.

- Port function registers (PFR0 to PFRH)
- Extended port function registers (EPFR0 to EPFR35)

This block enables analog output from pins for which output is enabled by the D/A control registers (DACR0 to DACR2). For details of the D/A control registers (DACR0 to DACR2), see "CHAPTER 27 8-bit D/A Converter".

<Note>

The D/A analog output multifunction ports are P92 to P90.

■ Clocks

Table 15.2-1 lists the clocks used for I/O ports.

Table 15.2-1 Clocks used for I/O ports

Clock name	Description
Operation clock	Peripheral clock (PCLK)

15.3 Pins

This section explains the pins of I/O ports.

■ Overview

Up to 99 I/O ports are provided, and they are categorized into port 0 to port K.

The I/O ports belonging to a port with the same suffix can be read/written at the same time.

- P00 to P07 (port 0)
- P10 to P17 (port 1)
- P20 to P27 (port 2)
- P30 to P37 (port 3)
- P40 to P47 (port 4)
- P50 to P57 (port 5)
- P60 to P67 (port 6)
- P70 to P77 (port 7)
- P80 to P87 (port 8)
- P90 to P92 (port 9)
- PA0 to PA7 (port A)
- PG0 to PG7 (port G)
- PH0 to PH3 (port H)
- PK0 to PK3 (port K)

15.4 Registers

This section explains the configuration and functions of the registers used for I/O ports.

■ List of registers for I/O ports

Table 15.4-1 lists the registers for I/O ports.

Table 15.4-1 Registers for I/O ports (1 / 2)

Port	Abbreviated Register Name	Register Name	Reference
Common	EPFR0 to EPFR35	Extended port function register 0 to 35	15.4.3
	ADCHE	A/D channel enable register	15.4.6
0	DDR0	Port data direction register 0	15.4.1
	PFR0	Port function register 0	15.4.2
	PCR0	Pull-up resistor control register 0	15.4.5
	PDR0	Port data register 0	15.4.4
1	DDR1	Port data direction register 1	15.4.1
	PFR1	Port function register 1	15.4.2
	PCR1	Pull-up resistor control register 1	15.4.5
	PDR1	Port data register 1	15.4.4
2	DDR2	Port data direction register 2	15.4.1
	PFR2	Port function register 2	15.4.2
	PDR2	Port data register 2	15.4.4
3	DDR3	Port data direction register 3	15.4.1
	PFR3	Port function register 3	15.4.2
	PDR3	Port data register 3	15.4.4
4	DDR4	Port data direction register 4	15.4.1
	PFR4	Port function register 4	15.4.2
	PDR4	Port data register 4	15.4.4
5	DDR5	Port data direction register 5	15.4.1
	PFR5	Port function register 5	15.4.2
	PCR5	Pull-up resistor control register 5	15.4.5
	PDR5	Port data register 5	15.4.4

Table 15.4-1 Registers for I/O ports (2 / 2)

Port	Abbreviated Register Name	Register Name	Reference
6	DDR6	Port data direction register 6	15.4.1
	PFR6	Port function register 6	15.4.2
	PCR6	Pull-up resistor control register 6	15.4.5
	PDR6	Port data register 6	15.4.4
7	DDR7	Port data direction register 7	15.4.1
	PFR7	Port function register 7	15.4.2
	PCR7	Pull-up resistor control register 7	15.4.5
	PDR7	Port data register 7	15.4.4
8	DDR8	Port data direction register 8	15.4.1
	PFR8	Port function register 8	15.4.2
	PCR8	Pull-up resistor control register 8	15.4.5
	PDR8	Port data register 8	15.4.4
9	DDR9	Port data direction register 9	15.4.1
	PCR9	Pull-up resistor control register 9	15.4.5
	PDR9	Port data register 9	15.4.4
A	DDRA	Port data direction register A	15.4.1
	PFRA	Port function register A	15.4.2
	PCRA	Pull-up resistor control register A	15.4.5
	PDRA	Port data register A	15.4.4
G	DDRG	Port data direction register G	15.4.1
	PFRG	Port function register G	15.4.2
	PDRG	Port data register G	15.4.4
H	DDRH	Port data direction register H	15.4.1
	PFRH	Port function register H	15.4.2
	PDRH	Port data register H	15.4.4
K	DDRK	Port data direction register K	15.4.1
	PCRK	Pull-up resistor control register K	15.4.5
	PDRK	Port data register K	15.4.4

15.4.1 Port Data Direction Registers (DDR0 to DDRK)

These registers set the I/O directions of pins used as general-purpose ports.
For a pin for a peripheral function or the external bus interface, these registers set the contents read from a port data register (PDR0 to PDRK).

The meaning of a read/written value of the port data register (PDR0 to PDRK) varies depending on the setting of each bit in this port data direction register and the settings of a port function register (PFR0 to PFRH).

Figure 15.4-1 shows the bit configuration of the port data direction registers (DDR0 to DDRK).

Figure 15.4-1 Bit configuration of the port data direction registers (DDR0 to DDRK)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
DDR0	DDR07	DDR06	DDR05	DDR04	DDR03	DDR02	DDR01	DDR00	0000 0000	R/W
DDR1	DDR17	DDR16	DDR15	DDR14	DDR13	DDR12	DDR11	DDR10	0000 0000	R/W
DDR2	DDR27	DDR26	DDR25	DDR24	DDR23	DDR22	DDR21	DDR20	0000 0000	R/W
DDR3	DDR37	DDR36	DDR35	DDR34	DDR33	DDR32	DDR31	DDR30	0000 0000	R/W
DDR4	DDR47	DDR46	DDR45	DDR44	DDR43	DDR42	DDR41	DDR40	0000 0000	R/W
DDR5	DDR57	DDR56	DDR55	DDR54	DDR53	DDR52	DDR51	DDR50	0000 0000	R/W
DDR6	DDR67	DDR66	DDR65	DDR64	DDR63	DDR62	DDR61	DDR60	0000 0000	R/W
DDR7	DDR77	DDR76	DDR75	DDR74	DDR73	DDR72	DDR71	DDR70	0000 0000	R/W
DDR8	DDR87	DDR86	DDR85	DDR84	DDR83	DDR82	DDR81	DDR80	0000 0000	R/W
DDR9	Undefined	Undefined	Undefined	Undefined	Undefined	DDR92	DDR91	DDR90	XXXX X000	R/W
DDRA	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0	0000 0000	R/W
DDRG	DDRG7	DDRG6	DDRG5	DDRG4	DDRG3	DDRG2	DDRG1	DDRG0	0000 0000	R/W
DDRH	Undefined	Undefined	Undefined	Undefined	DDRH3	DDRH2	DDRH1	DDRH0	XXXX 0000	R/W
DDRK	Undefined	Undefined	Undefined	Undefined	DDRK3	DDRK2	DDRK1	DDRK0	XXXX 0000	R/W

R/W: Read/Write

X: Undefined

Each bit sets the I/O direction of the corresponding port.

Written Value	Explanation
0	Input direction
1	Output direction

The meaning of a read/written value of a port data register (PDR0 to PDRK) varies depending on the setting of each bit in one of these port data direction registers and the settings of a port function register (PFR0 to PFRH).

Table 15.4-2 shows the relationship between the register settings and read/written values of the port data registers (PDR0 to PDRK).

Table 15.4-2 Relationship between register settings and read/written values of the port data registers (PDR0 to PDRK)

Mode	DDR	PFR	PDR	
Port input mode	0	0	In case of reading (except RMW instructions)	The output level of an external pin is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.
Port output mode	1	0	In case of reading (except RMW instructions)	The PDR value is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR and output to an external pin.
Peripheral function output mode *	0	1	In case of reading (except RMW instructions)	The output level from a peripheral function is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.
	1	1	In case of reading (except RMW instructions)	The PDR value is read.
			In case of reading (RMW instructions)	The PDR value is read.
			In case of writing	The written value is saved in a PDR.

* The functions of the output pins of external functions must be assigned to the appropriate pins by the extended port function registers (EPFR0 to EPFR35), and output from the pins must be enabled.

DDR: Port data direction register (DDR0 to DDRK)

PFR: Port function register (PFR0 to PFRH)

PDR: Port data register (PDR0 to PDRK)

RMW instruction: Read-modify-write instruction

<Notes>

- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR35). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter is enabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z. Furthermore, when output from the 8-bit D/A converter is enabled, the settings of each register are disabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z.

  In serial write mode selected by the MD1 and MD0 pins (MD1, MD0 = 01), digital input is enabled and analog input is disabled only for P75 (AN5 pin).
- When this device is reset, the settings of these registers are reset to the initial value (00_H), and the I/O direction of every port becomes input.
- To use PK0 and PK1 as low-speed oscillation pins, be sure to set the I/O directions of the ports to input (DDRK0 = 0, DDRK1 = 0) in port data direction register K (DDRK).

  (If PK0 and PK1 is used as a low-speed oscillation pin when the I/O direction of the related port has been set to output, the PDR value is output from the pin when low-speed oscillation is disabled.)

15.4.2 Port Function Registers (PFR0 to PFRH)

These registers select how to use individual pins.

The meaning of a read/written value of a port data register (PDR0 to PDRK) varies depending on the setting of each bit in one of these port function registers and the settings of a port data direction register (DDR0 to DDRK).

For details, see "15.4.1 Port Data Direction Registers (DDR0 to DDRK)".

Figure 15.4-2 shows the bit configuration of the port function registers (PFR0 to PFRH).

Figure 15.4-2 Bit configuration of the port function registers (PFR0 to PFRH)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PFR0	PFR07	PFR06	PFR05	PFR04	PFR03	PFR02	PFR01	PFR00	0000 0000	R/W
PFR1	PFR17	PFR16	PFR15	PFR14	PFR13	PFR12	PFR11	PFR10	0000 0000	R/W
PFR2	PFR27	PFR26	PFR25	PFR24	PFR23	PFR22	PFR21	PFR20	0000 0000	R/W
PFR3	PFR37	PFR36	PFR35	PFR34	PFR33	PFR32	PFR31	PFR30	0000 0000	R/W
PFR4	PFR47	PFR46	PFR45	PFR44	PFR43	PFR42	PFR41	PFR40	0000 0000	R/W
PFR5	PFR57	PFR56	PFR55	PFR54	PFR53	PFR52	PFR51	PFR50	0000 0000	R/W
PFR6	PFR67	PFR66	Undefined	PFR64	PFR63	Undefined	PFR61	Undefined	00X0 0X0X	R/W
PFR7	PFR77	PFR76	PFR75	PFR74	PFR73	PFR72	PFR71	PFR70	0000 0000	R/W
PFR8	PFR87	PFR86	PFR85	PFR84	PFR83	PFR82	PFR81	PFR80	0000 0000	R/W
PFRA	PFRA7	PFRA6	Undefined	PFRA4	PFRA3	PFRA2	PFRA1	PFRA0	00X0 0000	R/W
PFRG	Undefined	PFRG6	PFRG5	PFRG4	Undefined	PFRG2	PFRG1	PFRG0	X000 X000	R/W
PFRH	Undefined	Undefined	Undefined	Undefined	Undefined	PFRH2	Undefined	PFRH0	XXXX X0X0	R/W

R/W: Read/Write

X: Undefined

The port function registers specify each pin as either a pin used as a general-purpose port or a pin for the peripheral function specified by an extended port function register (EPFR0 to EPFR35).

Written Value	Explanation
0	General-purpose port
1	Peripheral function (external bus pin in external bus mode *)

* To use a pin as an external bus pin, disable output of the peripheral function of the related port through the settings of an extended port function register (EPFR0 to EPFR35), or assign the pin to another port.

The following function and I/O settings can be made for each pin according to the settings of bits in one of these registers and the corresponding bits in an extended port function register (EPFR0 to EPFR35):

- In single-chip mode

PFR	EPFR	Function of Corresponding Pin	Output from Peripheral Functions	Input to Peripheral Functions	Port Output
0	0	Port	Disabled	Enabled	-
1	Sets the function assigned to the output pin of a peripheral function.	Output pin of a peripheral function	Enabled	Enabled	Disabled
	Sets the function not assigned to the output pin of a peripheral function	Port	Disabled	Disabled	- (depending on DDR)

PFR: Corresponding bit in a port function register (PFR0 to PFRH)

EPFR: Corresponding bit in an extended port function register (EPFR0 to EPFR35)

- In external bus mode

PFR	EPFR	Function of Corresponding Pin	Output from Peripheral Functions	Input to Peripheral Functions	Port Output	External Bus Pin Output	External Bus Pin Input
0	0	Port	Disabled	Enabled	-	Disabled	Enabled
1	Sets the function assigned to the output pin of a peripheral function and enables output.	Output pin of a peripheral function	Enabled	Enabled	Disabled	Disabled	Enabled
	Cancels the assignment of a function to the output pin of a peripheral function, or disables output	External bus pin (only for multifunction ports for external bus pin functions)	Disabled	Enabled	Disabled	Enabled	Enabled
		Other than external bus pin	Disabled	Enabled (only the assigned function)	- (depending on DDR)	-	-

PFR: Corresponding bit in a port function register (PFR0 to PFRH)

EPFR: Corresponding bit in an extended port function register (EPFR0 to EPFR35)

<Notes>

- When this device is reset, the settings of these registers are reset to the initial value (00_H), and all ports are set to operate as input ports.
- If this register specifies a pin as a general-purpose port, the corresponding pin will operate as a general-purpose port even if a peripheral function has been assigned to that pin in one of the extended port function registers (EPFR0 to EPFR35).
- When analog input is enabled through the settings of the A/D channel enable register (ADCHE), input from ports and other functions is fixed at "0" regardless of the settings of these registers.

  When output from the 8-bit D/A converter is enabled by the DAE bit (DAE = 1) of the D/A control registers (DACR0 to DACR2), input from ports is fixed at "0" regardless of the settings of these registers. For details of the D/A control registers (DACR0 to DACR2), see "CHAPTER 27 8-bit D/A Converter".

  To enable the functions of external bus interface pins in external bus mode, make the following settings:

  1. Disable output from all peripheral functions by using the corresponding bits of the extended port function registers (EPFR0 to EPFR35).
  2. Write "1" to the corresponding bits in this register to set peripheral functions as the functions of the pins (external bus pins in external bus mode).
- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR35). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter is enabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z. Furthermore, when output from the 8-bit D/A converter is enabled, the settings of each register are disabled, input is always fixed at "0", and output from the port is always fixed at Hi-Z.

15.4.3 Extended Port Function Registers (EPFR0 to EPFR35)

These registers set the pin to which a function is assigned from among the multiple pins available for the function. Output from such pins is enabled/disabled according to the registers.

Figure 15.4-3 shows the bit configuration of the extended port function registers (EPFR0 to EPFR35).

Figure 15.4-3 Bit configuration of the extended port function registers (EPFR0 to EPFR35)

bit	7	6	5	4	3	2	1	0	Initial value
EPFR0	Undefined	Undefined	OUT1E2	OUT1E1	OUT1E0	OUT0E2	OUT0E1	OUT0E0	XX00 0000
EPFR1	Undefined	Undefined	OUT3E2	OUT3E1	OUT3E0	OUT2E2	OUT2E1	OUT2E0	XX00 0000
EPFR2	Undefined	Undefined	OUT5E2	OUT5E1	OUT5E0	OUT4E2	OUT4E1	OUT4E0	XX00 0000
EPFR3	Undefined	Undefined	OUT7E2	OUT7E1	OUT7E0	OUT6E2	OUT6E1	OUT6E0	XX00 0000
EPFR4	IN3E1	IN3E0	IN2E1	IN2E0	IN1E1	IN1E0	IN0E1	IN0E0	0000 0000
EPFR5	IN7E1	IN7E0	IN6E1	IN6E0	IN5E1	IN5E0	IN4E1	IN4E0	0000 0000
EPFR6	SOUT0E2	SOUT0E1	SOUT0E0	SCK0E2	SCK0E1	SCK0E0	SIN0E1	SIN0E0	0000 0000
EPFR7	Undefined	Undefined	Undefined	SOUT1E1	SOUT1E0	SCK1E1	SCK1E0	SIN1E	XXX0 0000
EPFR8	Undefined	Undefined	Undefined	SOUT2E1	SOUT2E0	SCK2E1	SCK2E0	SIN2E	XXX0 0000
EPFR9	Undefined	Undefined	Undefined	SOUT3E1	SOUT3E0	SCK3E1	SCK3E0	SIN3E	XXX0 0000
EPFR10	Undefined	Undefined	Undefined	SOUT4E1	SOUT4E0	SCK4E1	SCK4E0	SIN4E	XXX0 0000
EPFR11	Undefined	Undefined	Undefined	SOUT5E1	SOUT5E0	SCK5E1	SCK5E0	SIN5E	XXX0 0000
EPFR12	Undefined	Undefined	Undefined	SOUT6E1	SOUT6E0	SCK6E1	SCK6E0	SIN6E	XXX0 0000
EPFR13	Undefined	Undefined	Undefined	SOUT7E1	SOUT7E0	SCK7E1	SCK7E0	SIN7E	XXX0 0000
EPFR14	Undefined	Undefined	Undefined	SOUT8E1	SOUT8E0	SCK8E1	SCK8E0	SIN8E	XXX0 0000
EPFR15	Undefined	Undefined	Undefined	SOUT9E1	SOUT9E0	SCK9E1	SCK9E0	SIN9E	XXX0 0000
EPFR16	Undefined	Undefined	Undefined	SOUT10E1	SOUT10E0	SCK10E1	SCK10E0	SIN10E	XXX0 0000
EPFR17	Undefined	Undefined	Undefined	SOUT11E1	SOUT11E0	SCK11E1	SCK11E0	SIN11E	XXX0 0000
EPFR18	UDIN3E1	UDIN3E0	UDIN2E1	UDIN2E0	UDIN1E1	UDIN1E0	UDIN0E1	UDIN0E0	0000 0000
EPFR19	Undefined	Undefined	Undefined	Undefined	ADTRG0E2	ADTRG0E1	ADTRG0E0	XAE	XXXX 0001
EPFR20	Undefined	Undefined	TIOA1E1	TIOA1E0	TIOB1E	TIOA0E1	TIOA0E0	TIOB0E	XX00 0000
EPFR21	Undefined	Undefined	TIOA3E1	TIOA3E0	TIOB3E	TIOA2E1	TIOA2E0	TIOB2E	XX00 0000
EPFR22	Undefined	Undefined	TIOA5E1	TIOA5E0	TIOB5E	TIOA4E1	TIOA4E0	TIOB4E	XX00 0000
EPFR23	Undefined	Undefined	TIOA7E1	TIOA7E0	TIOB7E	TIOA6E1	TIOA6E0	TIOB6E	XX00 0000
EPFR24	Undefined	Undefined	TIOA9E1	TIOA9E0	TIOB9E	TIOA8E1	TIOA8E0	TIOB8E	XX00 0000
EPFR25	Undefined	Undefined	TIOA11E1	TIOA11E0	TIOB11E	TIOA10E1	TIOA10E0	TIOB10E	XX00 0000
EPFR26	Undefined	Undefined	TIOA13E1	TIOA13E0	TIOB13E	TIOA12E1	TIOA12E0	TIOB12E	XX00 0000
EPFR27	Undefined	Undefined	TIOA15E1	TIOA15E0	TIOB15E	TIOA14E1	TIOA14E0	TIOB14E	XX00 0000
EPFR28	INT7E	INT6E	INT5E	INT4E	INT3E	INT2E	INT1E	INT0E	0000 0000
EPFR29	INT15E	INT14E	INT13E	INT12E	INT11E	INT10E	INT9E	INT8E	0000 0000
EPFR30	Undefined	Undefined	Undefined	Undefined	INT19E	INT18E	INT17E	INT16E	XXXX 0000
EPFR31	Undefined	INT23E1	INT23E0	INT22E1	INT22E0	INT21E1	INT21E0	INT20E	X000 0000
EPFR32	INT31E	INT30E	INT29E	INT28E	INT27E	INT26E	INT25E	INT24E	0000 0000
EPFR33	Undefined	Undefined	TMO1E1	TMO1E0	TMI1E	TMO0E1	TMO0E0	TMI0E	XX00 0000
EPFR34	Undefined	TMO2E1	TMO2E0	TMI2E	FRCK1E1	FRCK1E0	FRCK0E1	FRCK0E0	X000 0000
EPFR35	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	SIFE1	SIFE0	XXXX XX00

Attribute: R/W (Read/Write) for all the bits

X: Undefined

<Notes>

- The pins that are specified as general-purpose ports in settings of the port function registers (PFR0 to PFRH) are treated as general-purpose I/O ports regardless of the settings of these registers.
- When analog input is enabled through the settings of the A/D channel enable register (ADCHE), input from ports is fixed at "0" regardless of the settings of these registers or port function registers (PFR0 to PFRH).
- When output from the 8-bit D/A converter is enabled by the DAE bit (DAE = 1) of the D/A control registers (DACR0 to DACR2), input from ports is fixed at "0" and output from ports is fixed at Hi-Z regardless of the settings of these registers or port function registers (PFR0 to PFRH).

  For details of the D/A control registers (DACR0 to DACR2), see "CHAPTER 27 8-bit D/A Converter".
- A single pin cannot be used as an output pin for multiple peripheral functions. Also, a single output function cannot be assigned to multiple pins.
- A single pin can be used as an input pin for multiple peripheral functions. However, a single input function cannot be assigned to multiple pins.
- If multiple functions are assigned to one pin, the order of priority is as follows:
  1. X0A/X1A
  2. Multifunction serial interface
  3. Base timer
  4. 16-bit reload timer
  5. 32-bit output compare
  6. Slave interface
- The input to a peripheral function is always connected to the pin assigned by an appropriate bit in an extended port function register (EPFR0 to EPFR35). Use port input mode for input to a peripheral function.

  However, when input from the 10-bit A/D converter or output from the 8-bit D/A converter is enabled, input is fixed at "0".
- Before changing the pin to which peripheral function output is assigned through the settings of this register, make the following settings:
  - Set port input mode for the pin to which the function is currently assigned and the pin to which it will be assigned.
  - Disable the assigned peripheral function.
- Before changing the pin to which a peripheral function input is assigned through the settings of this register, disable the assigned peripheral function.

● Extended port function register 0 (EPFR0) to extended port function register 3 (EPFR3)

[bit5 to bit0]: OUTxE2 to OUTxE0 (Output compare output pin select bits)

2 output pins for 32-bit output compare are provided for each channel.

These bits select the pins used by ch.0 to ch.7 for 32-bit output compare. The OUT0E2 to OUT0E0 bits correspond to ch.0, the OUT1E2 to OUT1E0 bits correspond to ch.1,..., and the OUT7E2 to OUT7E0 bits correspond to ch.7.

OUTxE2	OUTxE1	OUTxE0	Port Number	Pin Name
0	0	0	-	Output disabled
		1	Port 0	OUTx pin
	1	0	Port 1	OUTx_1 pin
		1	-	Setting prohibited
1	0	0	-	Setting prohibited
		1	-	Setting prohibited
	1	0	-	Setting prohibited
		1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus

● Extended port function register 4 (EPFR4) to extended port function register 5 (EPFR5)

[bit7 to bit0]: INxE1, INxE0 (Input capture input pin select bits)

3 input pins for 32-bit input capture are provided for each channel.

These bits select the pins used by ch.0 to ch.7 for 32-bit input capture. The IN0E1 and IN0E0 bits correspond to ch.0, the IN1E1 and IN1E0 bits correspond to ch.1,..., and the IN7E1 and IN7E0 bits correspond to ch.7.

INxE1	INxE0	Port Number	Pin Name
0	0	Port 0	INx pin
	1	Port 1	INx_1 pin
1	0	Port 2	INx_2 pin
	1	-	Setting prohibited

● Extended port function register 6 (EPFR6)

[bit7 to bit5]: SOUT0E2 to SOUT0E0 (Serial interface ch.0 serial data pin select bits)

These bits select one pin from the SOUT0, SOUT0_1, and SOUT0_2 pins to assign the serial data output function of multifunction serial interface ch.0 to the pin.

SOUT0E2	SOUT0E1	SOUT0E0	Port Number	Pin Name
0	0	0	-	Output disabled (Input: SOUT0 pin (Port 0))
		1	Port 0	SOUT0 pin
	1	0	Port 1	SOUT0_1 pin
		1	-	Setting prohibited
1	0	0	Port 2	SOUT0_2 pin
		1	-	Setting prohibited
	1	0	-	Setting prohibited
		1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins selected by the following bits must be assigned to the same port number:
  - SOUT0E2 to SOUT0E0 (serial data output pins)
  - SCK0E2 to SCK0E0 (serial clock I/O pins)
  - SIN0E1, SIN0E0 (serial data input pins)
- The serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "000", the input is connected to the SOUT0 pin (port 0).

[bit4 to bit2]: SCK0E2 to SCK0E0 (Serial interface ch.0 serial clock pin select bits)

These bits select one pin from the SCK0, SCK0_1, and SCK0_2 pins to assign the serial clock I/O function of multifunction serial interface ch.0 to the pin.

SCK0E2	SCK0E1	SCK0E0	Port Number	Pin Name
0	0	0	-	Output disabled (Input: SCK0 pin (Port 0))
		1	Port 0	SCK0 pin
	1	0	Port 1	SCK0_1 pin
		1	-	Setting prohibited
1	0	0	Port 2	SCK0_2 pin
		1	-	Setting prohibited
	1	0	-	Setting prohibited
		1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins selected by the following bits must be assigned to the same port number:
  - SOUT0E2 to SOUT0E0 (serial data output pins)
  - SCK0E2 to SCK0E0 (serial clock I/O pins)
  - SIN0E1, SIN0E0 (serial data input pins)
- The input of a peripheral function is always connected to the selected pin, and if these bits are set to "000", the input is connected to the SCK0 pin (port 0).

[bit1, bit0]: SIN0E1, SIN0E0 (Serial interface ch.0 serial data input select bits)

These bits select one pin from the SIN0, SIN0_1, and SIN0_2 pins to assign the serial data input function of multifunction serial interface ch.0 to the pin.

SIN0E1	SIN0E0	Port Number	Pin Name
0	0	Port 0	SIN0 pin
	1	Port 1	SIN0_1 pin
1	0	Port 2	SIN0_2 pin
	1	-	Setting prohibited

<Note>

The pins selected by the following bits must be assigned to the same port number:

- SOUT0E2 to SOUT0E0 (serial data output pins)
- SCK0E2 to SCK0E0 (serial clock I/O pins)
- SIN0E1, SIN0E0 (serial data input pins)

● Extended port function register 7 (EPFR7), extended port function register 8 (EPFR8)

[bit4, bit3]: SOUTxE1, SOUTxE0 (Serial interface ch.1, ch.2 serial data pin select bits)

2 serial data output pins are provided for each channel in multifunction serial interface ch.1, ch.2.

These bits select one of the pins to assign the serial data output function to it for each channel. The SOUT1E1 and SOUT1E0 bits correspond to ch.1, the SOUT2E1 and SOUT2E0 bits correspond to ch.2.

SOUTxE1	SOUTxE0	Port Number	Pin Name
0	0	-	Output disabled (Input: SOUTx pin (Port 0))
0	1	Port 0	SOUTx pin
1	0	Port 1	SOUTx_1 pin
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:
  - SOUTxE1, SOUTxE0 (serial data output pins)
  - SCKxE1, SCKxE0 (serial clock I/O pins)
  - SINxE (serial data input pin)
- Serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SOUTx pin (port 0).

[bit2, bit1]: SCKxE1, SCKxE0 (Serial interface ch.1, ch.2 serial clock pin select bits)

2 serial clock I/O pins are provided for each channel in multifunction serial interface ch.1, ch.2.

These bits select one of the pins to assign the serial clock I/O function to it for each channel. The SCK1E1 and SCK1E0 bits correspond to ch.1, the SCK2E1 and SCK2E0 bits correspond to ch.2.

SCKxE1	SCKxE0	Port Number	Pin Name
0	0	-	Output disabled (Input: SCKx pin (Port 0))
0	1	Port 0	SCKx pin
1	0	Port 1	SCKx_1 pin
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:
  - SOUTxE1, SOUTxE0 (serial data output pins)
  - SCKxE1, SCKxE0 (serial clock I/O pins)
  - SINxE (serial data input pin)
- The input of the serial clock is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SCKx pin (port 0).

[bit0]: SINxE (Serial interface ch.1, ch.2 serial data input select bits)

2 serial data input pins are provided for each channel in multifunction serial interface ch.1, ch.2.

These bits select one of the pins to assign the serial data input function to it for each channel. The SIN1E bit corresponds to ch.1, the SIN2E bit corresponds to ch.2.

SINxE	Port Number	Pin Name
0	Port 0	SINx pin
1	Port 1	SINx_1 pin

<Note>

The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:

- SOUT0E2 to SOUT0E0 (serial data output pins)
- SCK0E2 to SCK0E0 (serial clock I/O pins)
- SIN0E1, SIN0E0 (serial data input pins)

● Extended port function register 9 (EPFR9) to extended port function register 17 (EPFR17)

[bit4, bit3]: SOUTxE1, SOUTxE0 (Serial interface ch.3 to ch.11 serial data pin select bits)

These bits select whether to enable the serial data output pin of each channel in multifunction serial interface ch.3 to ch.11. The SOUT3E1 and SOUT3E0 bits correspond to ch.3, the SOUT4E1 and SOUT4E0 bits correspond to ch.4,... and the SOUT11E1 and SOUT11E0 bits correspond to ch.11.

SOUTxE1	SOUTxE0	Port Number	Pin name
0	0	-	Output disabled (Input: SOUTx pin (Port 0))
0	1	Port 0	SOUTx pin
1	0	-	Setting prohibited
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the pins are not used as output pins for other functions when output has been disabled by the bits, the pins can be used as follows:
  - In single chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- Serial data pins operate as input pins according to peripheral function settings. The input of a peripheral function is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SOUTx pin (port 0).

[bit2, bit1]: SCKxE1, SCKxE0 (Serial interface ch.3 to ch.11 serial clock pin select bits)

These bits select whether to enable the serial clock I/O pin of each channel in multifunction serial interface ch.3 to ch.11. The SCK3E1 and SCK3E0 bits correspond to ch.3, the SCK4E1 and SCK4E0 bits correspond to ch.4,... and the SCK11E1 and SCK11E0 bits correspond to ch.11.

SCKxE1	SCKxE0	Port Number	Pin name
0	0	-	Output disabled (Input: SCKx pin (Port 0))
0	1	Port 0	SCKx pin
1	0	-	Setting prohibited
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the pins are not used as output pins for other functions when output has been disabled by the bits, the pins can be used as follows:
  - In single chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The input of the serial clock is always connected to the selected pin, and if these bits are set to "00", the input is connected to the SCKx pin (port 0).

[bit0]: SINxE (Serial interface ch.3 to ch.11 serial data input select bit)

This bit selects one pin to assign it as the serial data input pin of multifunction serial interface ch.3 to ch.11. Always set "0" to this bit.

SINxE	Port Number	Pin name
0	Port 0	SINx pin
1	-	Setting prohibited

● Extended port function register 18 (EPFR18)

[bit1, bit0]: UDIN0E1, UDIN0E0 (Up/Down counter input pin select bits)

3 pins are provided for use by each channel in ch.0 of the 16-bit up/down counter.

These bits select one of the pins as the pin used in the 16-bit up/down counter of ch.0.

UDIN0E1	UDIN0E0	Port Number	Pin Name
0	0	Port 0	AIN0/BIN0/ZIN0 pins
0	1	Port 1	AIN0_1/BIN0_1/ZIN0_1 pins
1	0	Port 2	AIN0_2/BIN0_2/ZIN0_2 pins
1	1	-	Setting prohibited

[bit7 to bit2]: UDINxE1, UDINxE0 (Up/Down counter input pin select bits)

2 pins are provided for use by each channel in ch.1 to ch.3 of the 16-bit up/down counter.

These bits select one of the pins as the pin used by each channel in the 16-bit up/down counter. The UDIN1E1 and UDIN1E0 bits correspond to ch.1,..., and the UDIN3E1 and UDIN3E0 bits correspond to ch.3.

UDINxE1	UDINxE0	Port number	Pin name
0	0	Port 0	AINx/BINx/ZINx pins
0	1	Port 1	AINx_1/BINx_1/ZINx_1 pins
1	0	-	Setting prohibited
1	1	-	Setting prohibited

● Extended port function register 19 (EPFR19)

[bit3 to bit1]: ADTRG0E2 to ADTRG0E0 (A/D conversion activation trigger pin select bits)

4 external trigger input pins for 10-bit A/D converter are provided.

ADTRG0E2	ADTRG0E1	ADTRG0E0	Port Number	Pin Name
0	0	0	Port 0	ADTRG0 pin
		1	Port 1	ADTRG0_1 pin
	1	0	Port 2	ADTRG0_2 pin
		1	Port 3	ADTRG0_3 pin
1	0	0	-	Setting prohibited
		1	-	Setting prohibited
	1	0	-	Setting prohibited
		1	-	Setting prohibited

[bit0]: XAE (Clock oscillation I/O pin enable bit)

This bit cuts off port input when the low-speed clock oscillation function is enabled. Always set XAE = 1 when the low-speed clock oscillation function is enabled.

Written Value	Explanation
0	Enables port input.
1	Disables port input.

<Note>

These pins can be used as follows when the low-speed oscillation function has been disabled by this bit:

- In single-chip mode: General-purpose port
- In external bus mode (except external bus pins): General-purpose port
- In external bus mode: External bus

● Extended port function register 20 (EPFR20) to extended port function register 22 (EPFR22)

[bit5, bit4, bit2, bit1]: TIOAxE1, TIOAxE0 (Base timer ch.0 to ch.5 pin select bits)

2 output pins are provided for each channel in base timer ch.0 to ch.5.

These bits select one of the pins as the pin used by each channel in base timer ch.0 to ch.5. The TIOA0E1 and TIOA0E0 bits correspond to ch.0, the TIOA1E1 and TIOA1E0 bits correspond to ch.1,..., and the TIOA5E1 and TIOA5E0 bits correspond to ch.5.

TIOAxE1	TIOAxE0	Port Number	Pin Name
0	0	-	Output disabled (Odd-numbered channel input: TIOAx pin (Port 0))
0	1	Port 0	TIOAx pin
1	0	Port 1	TIOAx_1 pin
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:
  - TIOAxE1, TIOAxE0 (base timer output pins)
  - TIOBxE (base timer input pin)
- The base timer output pins (TIOAx pins) of the odd-numbered channels (TIOAx pin) operate as input pins according peripheral function settings. The input of a peripheral function is always connected to the selected pin. If these bits are set to "00", the input is connected to the TIOAx pin (port 0).

[bit3, bit0]: TIOBxE (Base timer ch.0 to ch.5 pin input select bits)

2 input pins are provided for each channel in base timer ch.0 to ch.5.

These bits select one of the pins as the pin used by each channel in base timer ch.0 to ch.5. The TIOB0E bit corresponds to ch.0, the TIOB1E bit corresponds to ch.1,..., and the TIOB5E bit corresponds to ch.5.

TIOBxE	Port Number	Pin Name
0	Port 0	TIOBx pin
1	Port 1	TIOBx_1 pin

<Note>

The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:

- TIOAxE1, TIOAxE0 (base timer output pins)
- TIOBxE (base timer input pin)

● Extended port function register 23 (EPFR23) to extended port function register 27 (EPFR27)

[bit5, bit4, bit2, bit1]: TIOAxE1, TIOAxE0 (Base timer ch.6 to ch.15 pin select bits)

These bits select whether to enable the output pin of each channel in base timer ch.6 to ch.15. The TIOA6E1 and TIOA6E0 bits correspond to ch.6, the TIOA7E1 and TIOA7E0 bits correspond to ch.7,... and the TIOA15E1 and TIOA15E0 bits correspond to ch.15.

TIOAxE1	TIOAxE0	Port Number	Pin name
0	0	-	Output disabled (Odd-numbered channel input: TIOAx pin (Port 0))
0	1	Port 0	TIOAx pin
1	0	-	Setting prohibited
1	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the pins are not used as output pins for other functions when output has been disabled by the bits, the pins can be used as follows:
  - In single chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The base timer output pins (TIOAx pins) of the odd-numbered channels (TIOAx pin) operate as input pins according peripheral function settings. The input of a peripheral function is always connected to the selected pin. If these bits are set to "00", the input is connected to the TIOAx pin (port 0).

[bit3, bit0]: TIOBxE (Base timer ch.6 to ch.15 pin input select bits)

These bits select whether to enable the input pin of each channel in base timer ch.6 to ch.15. The TIOB6E bit corresponds to ch.6, the TIOB7E bit corresponds to ch.7,... and the TIOB15E bit corresponds to ch.15.

TIOBxE	Port Number	Pin name
0	Port 0	TIOBx pin
1	-	Setting prohibited

● Extended port function register 28 (EPFR28)

[bit7 to bit4]: INT7E to INT4E (External interrupt request pin enable bits)

These bits select one of the pins as that used by each channel in external interrupt request ch.4 to ch.7. The INT7E bit corresponds to ch.7, the INT6E bit corresponds to ch.6,... and the INT4E bit corresponds to ch.4.

INTxE	Port Number	Pin name
0	Port 0	INTx pin
1	-	Setting prohibited

[bit3 to bit0]: INT3E to INT0E (External interrupt request pin enable bits)

Two input pins are provided for each channel in external interrupt request ch.0 to ch.3.

These bits select one of the pins as that used by each channel in external interrupt request ch.0 to ch.3. The INT0E bit corresponds to ch.0, the INT1E bit corresponds to ch.1,... and the INT3E bit corresponds to ch.3.

INTxE	Port number	Pin name
0	Port 0	INTx pin
1	Port 1	INTx_1 pin

● Extended port function register 29 (EPFR29)

[bit7 to bit0]: INT15E to INT8E (External interrupt request pin enable bits)

These bits select one of the pins as that used by each channel in external interrupt request ch.8 to ch.15.

The INT8E bit corresponds to ch.8, the INT9E bit corresponds to ch.9,... and the INT15E bit corresponds to ch.15.

INTxE	Port Number	Pin name
0	Port 0	INTx pin
1	-	Setting prohibited

● Extended port function register 30 (EPFR30)

[bit3 to bit0]: INT19E to INT16E (External interrupt request pin enable bits)

2 input pins are provided for each channel in external interrupt request ch.16 to ch.19.

These bits select one of the pins as the pin used by each channel in external interrupt request ch.16 to ch.19. The INT16E bit corresponds to ch.16, the INT17E bit corresponds to ch.17,..., and the INT19E bit corresponds to ch.19.

INTxE	Port Number	Pin Name
0	Port 0	INTx pin
1	Port 1	INTx_1 pin

● Extended port function register 31 (EPFR31)

[bit6 to bit1]: INT23E1, INT23E0 to INT21E1, INT21E0 (External interrupt request pin enable bits)

3 input pins are provided for each channel in external interrupt request ch.21 to ch.23.

These bits select one of the pins as the pin used by each channel in external interrupt request ch.21 to ch.23. The INT21E1 and INT21E0 bits correspond to ch.21, the INT22E1 and INT22E0 bits correspond to ch.22, and the INT23E1 and INT23E0 bits correspond to ch.23.

INTxE1	INTxE0	Port Number	Pin Name
0	0	Port 0	INTx pin
0	1	Port 1	INTx_1 pin
1	0	Port 2	INTx_2 pin
1	1	-	Setting prohibited

[bit0]: INT20E (External interrupt request pin enable bit)

2 input pins are provided for external interrupt request ch.20.

This bit selects one of the pins as the pin used by external interrupt request ch.20.

INT20E	Port Number	Pin Name
0	Port 0	INT20 pin
1	Port 1	INT20_1 pin

● Extended port function register 32 (EPFR32)

[bit7 to bit0]: INT31E to INT24E (External interrupt request pin enable bits)

These bits select one of the pins as that used by each channel in external interrupt request ch.24 to ch.31.

The INT24E bit corresponds to ch.24, the INT25E bit corresponds to ch.25, ... and the INT31E bit corresponds to ch.31.

INTxE	Port Number	Pin Name
0	Port 0	INTx pin
1	-	Setting prohibited

● Extended port function register 33 (EPFR33)

[bit5, bit4, bit2, bit1]: TMOxE1, TMOxE0 (Reload timer ch.0 to ch.1 output pin select bits)

2 output pins are provided for each channel in 16-bit reload timer ch.0 and ch.1.

These bits select one of the pins as the pin used by each of 16-bit reload timer ch.0 and ch.1. The TMO0E1 and TMO0E0 bits correspond to ch.0, and the TMO1E1 and TMO1E0 bits correspond to ch.1.

TMOxE1	TMOxE0	Port Number	Pin Name
0	0	-	Output disabled
	1	Port 0	TMOx pin
1	0	Port 1	TMOx_1 pin
	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:
  - TMOxE1, TMOxE0 (16-bit reload timer output pins)
  - TMIxE (16-bit reload timer input pin)

[bit3, bit0]: TMIxE (Reload timer ch.0 to ch.1 input pin select bits)

2 input pins are provided for each channel in 16-bit reload timer ch.0 and ch.1.

These bits select one of the pins as the pin used by each of 16-bit reload timer ch.0 and ch.1. The TMI0E bit corresponds to ch.0, and the TMI1E bit corresponds to ch.1.

TMIxE	Port Number	Pin Name
0	Port 0	TMIx pin
1	Port 1	TMIx_1 pin

<Note>

The pins used for the same channel (the pins selected by the following bits) must be assigned to the same port number:

- TMOxE1, TMOxE0 (16-bit reload timer output pins)
- TMIxE (16-bit reload timer input pin)

● Extended port function register 34 (EPFR34)

[bit6, bit5]: TMO2E1, TMO2E0 (Reload timer ch.2 output pin select bits)

2 output pins are provided for 16-bit reload timer ch.2.

This bit selects one of the pins as the pin used by 16-bit reload timer ch.2.

TMO2E1	TMO2E0	Port Number	Pin Name
0	0	-	Output disabled
	1	Port 0	TMO2 pin
1	0	Port 1	TMO2_1 pin
	1	-	Setting prohibited

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- The pins selected by the following bits must be assigned to the same port number:
  - TMO2E1, TMO2E0 (16-bit reload timer output pins)
  - TMI2E (16-bit reload timer I/O pin)

[bit4]: TMI2E (Reload timer ch.2 input pin select bit)

2 input pins are provided for 16-bit reload timer ch.2.

This bit selects one of the pins as the pin used by 16-bit reload timer ch.2.

TMI2E	Port Number	Pin Name
0	Port 0	TMIx pin
1	Port 1	TMIx_1 pin

<Note>

The pins selected by the following bits must be assigned to the same port number:

- TMO2E1, TMO2E0 (16-bit reload timer output pins)
- TMI2E (16-bit reload timer I/O pin)

[bit3 to bit0]: FRCKxE1, FRCKxE0 (Free-run timer ch.0 and ch.1 input pin select bits)

2 input pins are provided for each channel in 32-bit free-run timer ch.0 and ch.1.

These bits select one of the pins as the pin used by each of 32-bit free-run timer ch.0 and ch.1.

FRCKxE1	FRCKxE0	Port Number	Pin Name
0	0	Port 0	FRCKx pin
	1	Port 1	FRCKx_1 pin
1	0	-	Setting prohibited
	1	-	Setting prohibited

● Extended port function register 35 (EPFR35)

[bit1, bit0]: SIFE1, SIFE0 (Slave interface output pin select bits)

Two slave interface pins are provided.

The functions of pins for the slave interfaces should be assigned and set the pins for the output to be enabled.

SIFE1	SIFE0	Port number	Pin name						
			Data I/O	Chip select input	Address input	Read strobe input	Write strobe input	Received data request output	Transmit data request output
0	0	-	output disabled (data input: 0 port, chip select input: 1 (fixed), address input 0 (fixed), write strobe input: 1 (fixed), read strobe input: 1 (fixed))						
0	1	Port 0	SIFD0 to SIFD15	$\overline{\text{SIFCS}}$	SIFA	$\overline{\text{SIFRD}}$	$\overline{\text{SIFWR}}$	$\overline{\text{SIFDRQR}}$	$\overline{\text{SIFDRQT}}$
1	0	Port 1	SIFD0_1 to SIFD15_1	$\overline{\text{SIFCS\_1}}$	SIFA_1	$\overline{\text{SIFRD\_1}}$	$\overline{\text{SIFWR\_1}}$	$\overline{\text{SIFDRQR\_1}}$	$\overline{\text{SIFDRQT\_1}}$
1	1	-	Setting prohibited						

<Notes>

- The corresponding pins can be used as output pins for other functions when output has been disabled by these bits.
- If the corresponding pins are not used as output pins for other functions when output has been disabled by these bits, the pins can be used as follows:
  - In single-chip mode: General-purpose port
  - In external bus mode (except external bus pins): General-purpose port
  - In external bus mode: External bus
- Slave interface input pins (SIFD0 to SIFD15, $\overline{\text{SIFCS}}$, SIFA, $\overline{\text{SIFRD}}$, $\overline{\text{SIFWR}}$) are always connected to the selected pins, and if these bits are set to "00", the input is connected to the port 0.
- If slave interface pins are enabled (SIFE1 and SIFE0 bit are "01" or "10") and need to change the assignment of the pin functions, disable the output before the change.

15.4.4 Port Data Registers (PDR0 to PDRK)

These registers store I/O data.

The values read from or written to these registers vary depending on the settings of a port data direction register (DDR0 to DDRK) and port function register (PFR0 to PFRH). For details of a read value or written value, see "15.4.1 Port Data Direction Registers (DDR0 to DDRK)".

Figure 15.4-4 shows the bit configuration of the port data registers (PDR0 to PDRK).

Figure 15.4-4 Bit configuration of the port data registers (PDR0 to PDRK)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PDR0	PDR07	PDR06	PDR05	PDR04	PDR03	PDR02	PDR01	PDR00	XXXX XXXX	R/W
PDR1	PDR17	PDR16	PDR15	PDR14	PDR13	PDR12	PDR11	PDR10	XXXX XXXX	R/W
PDR2	PDR27	PDR26	PDR25	PDR24	PDR23	PDR22	PDR21	PDR20	XXXX XXXX	R/W
PDR3	PDR37	PDR36	PDR35	PDR34	PDR33	PDR32	PDR31	PDR30	XXXX XXXX	R/W
PDR4	PDR47	PDR46	PDR45	PDR44	PDR43	PDR42	PDR41	PDR40	XXXX XXXX	R/W
PDR5	PDR57	PDR56	PDR55	PDR54	PDR53	PDR52	PDR51	PDR50	XXXX XXXX	R/W
PDR6	PDR67	PDR66	PDR65	PDR64	PDR63	PDR62	PDR61	PDR60	XXXX XXXX	R/W
PDR7	PDR77	PDR76	PDR75	PDR74	PDR73	PDR72	PDR71	PDR70	XXXX XXXX	R/W
PDR8	PDR87	PDR86	PDR85	PDR84	PDR83	PDR82	PDR81	PDR80	XXXX XXXX	R/W
PDR9	Undefined	Undefined	Undefined	Undefined	Undefined	PDR92	PDR91	PDR90	XXXX XXXX	R/W
PDRA	PDRA7	PDRA6	PDRA5	PDRA4	PDRA3	PDRA2	PDRA1	PDRA0	XXXX XXXX	R/W
PDRG	PDRG7	PDRG6	PDRG5	PDRG4	PDRG3	PDRG2	PDRG1	PDRG0	XXXX XXXX	R/W
PDRH	Undefined	Undefined	Undefined	Undefined	PDRH3	PDRH2	PDRH1	PDRH0	XXXX XXXX	R/W
PDRK	Undefined	Undefined	Undefined	Undefined	PDRK3	PDRK2	PDRK1	PDRK0	XXXX XXXX	R/W

R/W: Read/Write

X: Undefined

<Notes>

- If these registers are read with a read-modify-write instruction, the value of these registers is read regardless of the settings of the following registers:
  - Port data direction registers (DDR0 to DDRK)
  - Port function registers (PFR0 to PFRH)
- The value of these registers is not initialized even when this device is reset.

15.4.5 Pull-up Resistor Control Registers (PCR0 to PCRK)

These registers set pull-up resistors. One bit is provided for each of the pins for which pull-up resistors can be set, and a pull-up resistor can be set in the corresponding pin by writing "1" to the bit corresponding to the pin.

Figure 15.4-5 shows the bit configuration of the pull-up resistor control registers (PCR0 to PCRK).

Figure 15.4-5 Bit configuration of the pull-up resistor control registers (PCR0 to PCRK)

bit	7	6	5	4	3	2	1	0	Initial value	Attribute
PCR0	PCR07	PCR06	PCR05	PCR04	PCR03	PCR02	PCR01	PCR00	0000 0000	R/W
PCR1	PCR17	PCR16	PCR15	PCR14	PCR13	PCR12	PCR11	PCR10	0000 0000	R/W
PCR5	PCR57	PCR56	PCR55	PCR54	PCR53	PCR52	PCR51	PCR50	0000 0000	R/W
PCR6	PCR67	PCR66	PCR65	PCR64	PCR63	PCR62	PCR61	PCR60	0000 0000	R/W
PCR7	PCR77	PCR76	PCR75	PCR74	PCR73	PCR72	PCR71	PCR70	0000 0000	R/W
PCR8	PCR87	PCR86	PCR85	PCR84	PCR83	PCR82	PCR81	PCR80	0000 0000	R/W
PCR9	Undefined	Undefined	Undefined	Undefined	Undefined	PCR92	PCR91	PCR90	XXXX X000	R/W
PCRA	PCRA7	PCRA6	PCRA5	PCRA4	PCRA3	PCRA2	PCRA1	PCRA0	0000 0000	R/W
PCRK	Undefined	Undefined	Undefined	Undefined	PCRK3	PCRK2	Undefined	Undefined	XXXX 00XX	R/W

R/W: Read/Write

X: Undefined

Each bit in the pull-up resistor control registers specifies whether a pull-up resistor is set for the assigned pin.

When a pull-up this register is set, the pull-up resistor is connected to the pin.

Written Value	Explanation
0	The pull-up resistor is not set.
1	The pull-up resistor is set.

<Note>

Pull-up resistors are not set in the following cases regardless of the settings of these registers:

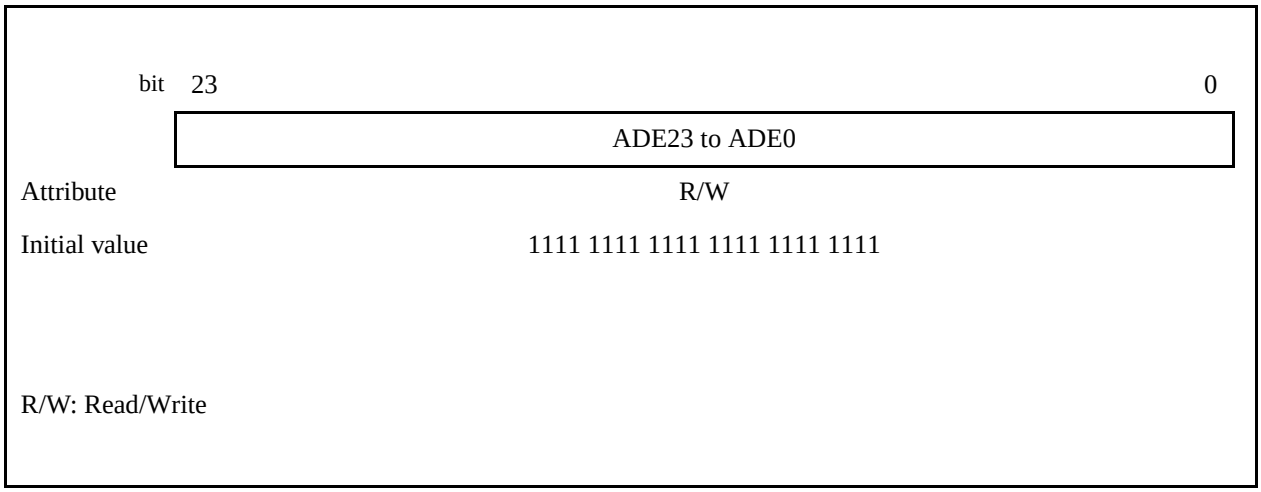
- In port output (in peripheral function output)
- In stop mode (with Hi-Z selected)
- When D/A analog output is enabled (only for port 9)

15.4.6 A/D Channel Enable Register (ADCHE)

This register specifies whether to input analog signals from the AN0 to AN23 pins.
One bit is provided for each of the pins for which A/D analog input can be set, and A/D analog input can be enabled for the corresponding pin by writing "1" to the bit corresponding to the pin.

Figure 15.4-6 shows the bit configuration of the A/D channel enable register (ADCHE).

Figure 15.4-6 Bit configuration of the A/D channel enable register (ADCHE)



[bit23 to bit0]: ADE23 to ADE0 (Analog input enable bits)

These bits enables/disables analog signal input from the pin corresponding to the bit.

Written Value	Explanation
0	Disables analog signal input.
1	Enables analog signal input.

The ADE23 bit corresponds to ch.23, the ADE22 bit corresponds to ch.22, the ADE21 bit corresponds to ch.21, ..., the ADE1 bit corresponds to ch.1, and the ADE0 bit corresponds to ch.0.

<Notes>

- To use any of the AN0 to AN23 pins as analog signal input pins of the 10-bit A/D converter, be sure to write "1" to the bits corresponding to the channels.
- When analog input is enabled through the settings of this register, input from ports and peripheral functions is fixed at "0" and output to them is fixed at Hi-Z regardless of the settings of the port function registers (PFR0 to PFRH) or extended port function registers (EPFR0 to EPFR35).

15.5 Notes on Use

Note the following points about using I/O ports:

- The order of priority of registers/signals is as follows:
  1. A/D channel enable register (ADCHE), D/A control registers (DACR0 to DACR2)
  2. Port function registers (PFR0 to PFRH)
  3. Extended port function registers (EPFR0 to EPFR35)
  4. External bus valid signals

  If settings are inconsistent, the setting with the higher order of priority is used.
- When output from the 8-bit D/A converter is enabled by the DAE bit (DAE = 1) of the D/A control registers (DACR0 to DACR2), input from ports is fixed at "0" and output from ports is fixed at Hi-Z.

  For details of the D/A control registers (DACR0 to DACR2), see "CHAPTER 27 8-bit D/A Converter".
- When analog input is enabled by the A/D channel enable register (ADCHE), input from ports is fixed at "0" and output from ports is fixed at Hi-Z.
- If multiple functions are assigned to one pin, the order of priority is as follows:
  1. X0A/X1A
  2. Multifunction serial interface
  3. Base timer
  4. 16-bit reload timer
  5. 32-bit output compare
  6. Slave interface
- A single pin cannot be used as an output pin for multiple peripheral functions. Also, a single output function cannot be assigned to multiple pins.
- A single pin can be used as an input pin for multiple peripheral functions. However, a single input function cannot be assigned to multiple pins.
- If Hi-Z is set to a pin in standby mode (stop mode/watch mode), input is fixed at "0". However, input is not fixed at "0" for external interrupt requests whose generation is enabled and it can be used.
- Before changing the pin to which a peripheral function output is assigned, set port input mode for the relevant pins (the pin to which the function is currently assigned and the pin to which it will be assigned) and disable the assigned peripheral function.
- Before changing the pin to which a peripheral function input is assigned, disable the assigned peripheral function.
- To use PK0 and PK1 as low-speed oscillation pins, set the I/O directions of the ports to input (DDRK0 = 0, DDRK1 = 0) in port data direction register K (DDRK).
- The pin to which peripheral functions are assigned can be set, if the peripheral functions can be assigned to more than one pin, and peripheral function output from the pin can be enabled/disabled.

  However, if the peripheral function has more than one I/O, each I/O must be set to individual ports belonging to the same group.

Example: Ch.1 multifunction serial interface settings

Serial Data Output	Serial Clock I/O	Serial Data Input	Effective port
SOUT1 pin (Port 0)	SCK1 pin (Port 0)	SIN1 pin (Port 0)	Port 0
		SIN1_1 pin (Port 1)	Setting prohibited
	SCK1_1 pin (Port 1)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	
SOUT1_1 pin (Port 1)	SCK1 pin (Port 0)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	
	SCK1_1 pin (Port 1)	SIN1 pin (Port 0)	
		SIN1_1 pin (Port 1)	Port 1

CHAPTER 16 External Interrupt Controllers

This chapter explains the functions and operations of external interrupt controllers.

16.1 Overview
16.2 Configuration
16.3 Pins
16.4 Registers
16.5 Explanation of Operations and Setting Procedure Examples

16.1 Overview

The external interrupt controllers detect edges/levels in external interrupt signals, and they control external interrupt requests.
This series has 32 built-in signal input pins for external interrupts.

■ Overview

An external interrupt controller generates an external interrupt request when it detects a preset edge/level in an external interrupt signal.

The edge/level to be detected can be selected from the following 4 types:

- "H" level
- "L" level
- Rising edge
- Falling edge

Also, external interrupt requests can be used for a return from sleep mode or standby mode (watch mode or stop mode).

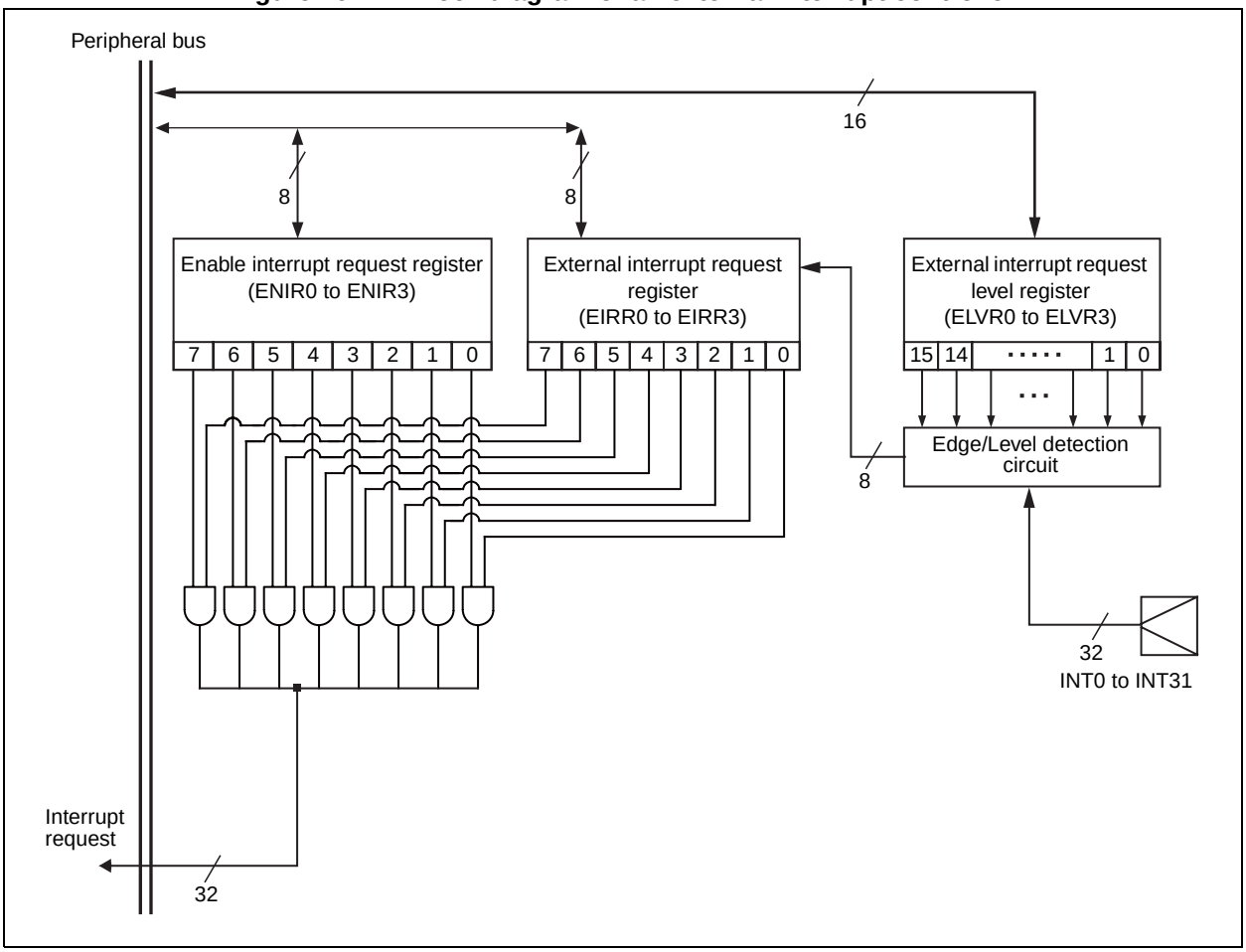
16.2 Configuration

This section shows the configuration of an external interrupt controller.

■ Block diagram of an external interrupt controller

Figure 16.2-1 is a block diagram of an external interrupt controller.

Figure 16.2-1 Block diagram of an external interrupt controller



- External interrupt request level register (ELVR0 to ELVR3)

  This register sets the edge/level used to determine whether a signal input to the INT0 to INT31 pins is for an external interrupt request.
- External interrupt request register (EIRR0 to EIRR3)

  This register maintains the states of interrupt sources (indicating which pins have generated external interrupt requests).
- Enable interrupt request register (ENIR0 to ENIR3)

  This register specifies whether external interrupt requests are enabled/disabled.
- Edge/Level detection circuit

  This circuit detects edges/levels in signals input to the INT0 to INT31 pins.

■ Clocks

Table 16.2-1 lists the clock used by the external interrupt controllers.

Table 16.2-1 Clock used by the external interrupt controllers

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

16.3 Pins

This section explains the pins of the external interrupt controllers.

■ Overview

The external interrupt controllers have the following pins:

- INT0 to INT31 pins

  These are external interrupt signal input pins.

  These pins are multiplexed pins. For details of using the INT0 to INT31 pins of the external interrupt controllers, see "2.4 Setting Method for Pins".

16.4 Registers

This section explains the configurations and functions of the registers for the external interrupt controllers.

■ List of registers for the external interrupt controllers

Table 16.4-1 lists the registers for the external interrupt controllers.

Table 16.4-1 Registers for the external interrupt controllers

Channel	Abbreviated Register Name	Register Name	Reference
Common	ELVR0	External interrupt request level register 0	16.4.1
	EIRR0	External interrupt request register 0	16.4.2
	ENIR0	Enable interrupt request register 0	16.4.3
	ELVR1	External interrupt request level register 1	16.4.1
	EIRR1	External interrupt request register 1	16.4.2
	ENIR1	Enable interrupt request register 1	16.4.3
	ELVR2	External interrupt request level register 2	16.4.1
	EIRR2	External interrupt request register 2	16.4.2
	ENIR2	Enable interrupt request register 2	16.4.3
	ELVR3	External interrupt request level register 3	16.4.1
	EIRR3	External interrupt request register 3	16.4.2
	ENIR3	Enable interrupt request register 3	16.4.3

16.4.1 External Interrupt Request Level Registers (ELVR0 to ELVR3)

These registers set the edges/levels to be detected for external interrupt requests.

Figure 16.4-1 shows the bit configuration of the external interrupt request level registers (ELVR0 to ELVR3).

Figure 16.4-1 Bit configuration of the external interrupt request level registers (ELVR0 to ELVR3)

External interrupt request level register 0 (ELVR0)

bit	15	14	13	12	11	10	9	8
	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request level register 1 (ELVR1)

bit	15	14	13	12	11	10	9	8
	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request level register 2 (ELVR2)

bit	15	14	13	12	11	10	9	8
	LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request level register 3 (ELVR3)

bit	15	14	13	12	11	10	9	8
	LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit15 to bit0]: LB31 to LB0, LA31 to LA0 (Detection condition selection bits)

These bits select the edges/levels to be detected in signals for external interrupt requests. An external interrupt request is recognized upon detection of the edge/level selected by one of these bits.

The LB31 to LB0 bits correspond to the INT0 to INT31 bits, and the LA31 to LA0 bits similarly correspond to the INT0 to INT31 bits. For example, the INT0 pin is set with the LB0 and LA0 bits.

LB31 to LB0	LA31 to LA0	Explanation
0	0	"L" level detection
0	1	"H" level detection
1	0	Rising edge detection
1	1	Falling edge detection

To use an external interrupt request to return from standby mode, see "16.5.2 Return from Standby Mode".

<Notes>

- For detection of an edge/level specified by these bits, the pulse width of the signal must be 4T or higher (T: Peripheral clock (PCLK) period). If a signal with a narrower pulse width is input, the MB91640/645 may not operate correctly.
- While "L" level detection/"H" level detection is set as the detection condition, the state of an interrupt source is maintained in the external interrupt request registers (EIRR0 to EIRR3) even if the corresponding external interrupt request is canceled. Therefore, the external interrupt request remains at the interrupt controller, to which it has been output. To cancel the external interrupt request output to the interrupt controller, set "0" in the corresponding bit in the external interrupt request register (EIRR0 to EIRR3).

  However, even when the external interrupt request register (EIRR0 to EIRR3) is cleared, the external interrupt request remains as is while any signals at the effective level are input from the INT0 to INT31 pins.

  For diagrams illustrating operations that maintain the state of an interrupt source or clear an interrupt source, see "■ Canceling an external interrupt request" of "16.5 Explanation of Operations and Setting Procedure Examples".
- If the detection condition is changed by rewriting these bits, an incorrect interrupt source may be generated. To prevent incorrect interrupt sources from being generated when the detection condition has been changed, perform the following operations:
  1. Read the external interrupt request level register (ELVR0 to ELVR3).
  2. Write "0" in the external interrupt request register (EIRR0 to EIRR3) to clear the interrupt source.

16.4.2 External Interrupt Request Registers (EIRR0 to EIRR3)

These registers maintain the states of interrupt sources of external interrupt requests (indicating which pins have generated the external interrupt requests).

Figure 16.4-2 shows the bit configuration of the external interrupt request registers (EIRR0 to EIRR3).

Figure 16.4-2 Bit configuration of the external interrupt request registers (EIRR0 to EIRR3)

External interrupt request register 0 (EIRR0)

bit	7	6	5	4	3	2	1	0
	ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request register 1 (EIRR1)

bit	7	6	5	4	3	2	1	0
	ER15	ER14	ER13	ER12	ER11	ER10	ER9	ER8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request register 2 (EIRR2)

bit	7	6	5	4	3	2	1	0
	ER23	ER22	ER21	ER20	ER19	ER18	ER17	ER16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

External interrupt request register 3 (EIRR3)

bit	7	6	5	4	3	2	1	0
	ER31	ER30	ER29	ER28	ER27	ER26	ER25	ER24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7 to bit0]: ER31 to ER0 (External interrupt request flag bits)

These bits indicate that external interrupt requests have been detected.

The ER0 to ER31 bits correspond to the INT0 to INT31 pins. For example, the ER0 bit is used to detect external interrupt requests from the INT0 pin, and the ER31 bit is used to detect external interrupt requests from the INT31 pin.

An external interrupt request is generated when "1" is set in any of the EN0 to EN31 bits of an enable interrupt request register (ENIR0 to ENIR3) and the corresponding bit among the ER0 to ER31 bits becomes "1".

ER31 to ER0	In Case of Reading	In Case of Writing
0	No external interrupt request has been detected.	The interrupt source is cleared.
1	An external interrupt request has been detected.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- As long as a signal at the effective level is being input from any of the INT0 to INT 31 pins when "L" level detection/"H" level detection has been set as the detection condition by an external interrupt request level register (ELVR0 to ELVR3), "1" is set in the corresponding bit among the ER31 to ER0 bits even after the bit is cleared.

16.4.3 Enable Interrupt Request Registers (ENIR0 to ENIR3)

These registers enable/disable external interrupt requests.

Figure 16.4-3 shows the bit configuration of the enable interrupt request registers (ENIR0 to ENIR3).

Figure 16.4-3 Bit configuration of the enable interrupt request registers (ENIR0 to ENIR3)

Enable interrupt request register 0 (ENIR0)

bit	7	6	5	4	3	2	1	0
	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Enable interrupt request register 1 (ENIR1)

bit	7	6	5	4	3	2	1	0
	EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Enable interrupt request register 2 (ENIR2)

bit	7	6	5	4	3	2	1	0
	EN23	EN22	EN21	EN20	EN19	EN18	EN17	EN16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Enable interrupt request register 3 (ENIR3)

bit	7	6	5	4	3	2	1	0
	EN31	EN30	EN29	EN28	EN27	EN26	EN25	EN24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7 to bit0]: EN31 to EN0 (Interrupt enable bits)

These bits enable/disable external interrupts.

Each of the EN0 to EN31 bits corresponds to the respective bits of the external interrupt request registers (EIRR0 to EIRR3).

Written Value	Explanation
0	Disables generation of external interrupt requests. The states of interrupt sources are maintained, but external interrupt requests are not output.
1	Enables generation of external interrupt requests. External interrupt requests are output.

16.5 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the external interrupt controllers and provides examples of setting procedures.

16.5.1 Operations of the External Interrupt Controllers

■ Overview

If external interrupts are enabled, an external interrupt controller outputs an external interrupt request when it detects a preset edge/level in a signal input to an external signal input pin.

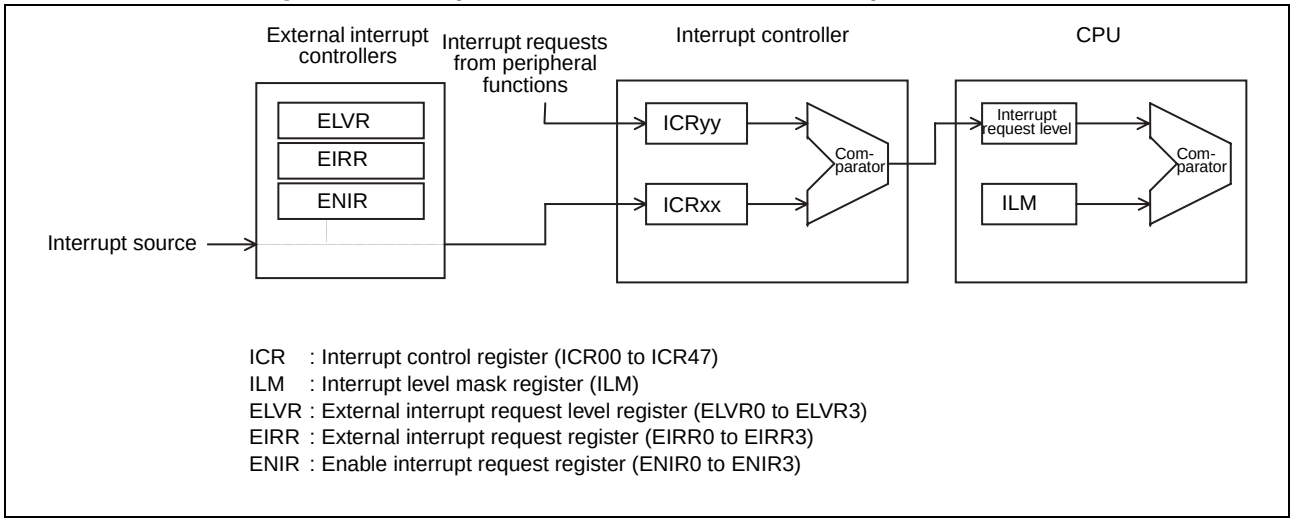
The edge/level to be detected can be selected from the following 4 types:

- "H" level
- "L" level
- Rising edge (Only when return from standby mode "L" level detection at the INT0 to INT7 pins, and rising edge detection at the INT8 to INT31 pins)
- Falling edge (Only when return from standby mode "H" level detection at the INT0 to INT7 pins, and falling edge detection at the INT8 to INT31 pins)

If an interrupt request from another peripheral device is generated at the same time, the interrupt controller determines their order of priority. An external interrupt is generated for the external interrupt request that has the higher priority.

Figure 16.5-1 shows operation with the external interrupt controllers.

Figure 16.5-1 Operation with the external interrupt controllers



■ Setting procedure

To set an external interrupt, follow the procedure below.

1. Disable external interrupts by using an enable interrupt request register (ENIR0 to ENIR3).
2. Change the detection condition (effective edge /level) by using an external interrupt request level register (ELVR0 to ELVR3).
3. Read the external interrupt request level register (ELVR0 to ELVR3).
4. Clear interrupt sources by using an external interrupt request register (EIRR0 to EIRR3).
5. Enable external interrupts by using the enable interrupt request register (ENIR0 to ENIR3).

<Notes>

- Before making settings for the external interrupt controller, disable external interrupts by using an enable interrupt request register (ENIR0 to ENIR3).
- Before enabling output of external interrupt requests, clear interrupt sources by using an external interrupt request register (EIRR0 to EIRR3).

■ Control operations

Each external interrupt controller issues external interrupt requests to the interrupt controller in the following sequence:

1. The external interrupt controller detects the edge/level specified by an external interrupt request level register (ELVR0 to ELVR3) in a signal input to any of the INT0 to INT31 pins.
2. The external interrupt controller determines whether external interrupts are enabled by checking the enable interrupt request registers (ENIR0 to ENIR3).
3. If external interrupts are enabled, the external interrupt controller outputs an external interrupt request to the interrupt controller.

■ Canceling an external interrupt request

While "L" level detection/"H" level detection is set as the detection condition for external interrupts, the state of an interrupt source is maintained in the external interrupt request registers (EIRR0 to EIRR3) even if the corresponding external interrupt request is canceled. Therefore, the external interrupt remains at the interrupt controller, to which a request for it has been output.

To cancel the external interrupt request output to the interrupt controller, set "0" in the corresponding bit in an external interrupt request register (EIRR0 to EIRR3). This operation clears the interrupt source, and the external interrupt request is canceled.

However, even when the external interrupt request register (EIRR0 to EIRR3) is cleared, the external interrupt remains at the interrupt controller, to which for a request it has been output, while any signals at the effective level are input from the INT0 to INT31 pins.

Figure 16.5-2 shows the state of an interrupt source being maintained, and Figure 16.5-3 shows the clearing of an interrupt source.

Figure 16.5-2 Maintaining the state of an interrupt source

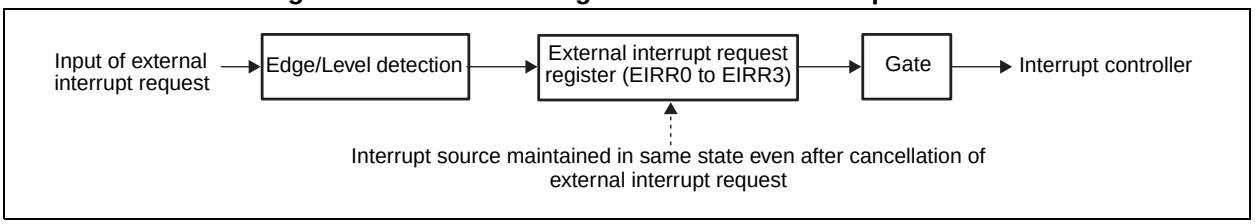
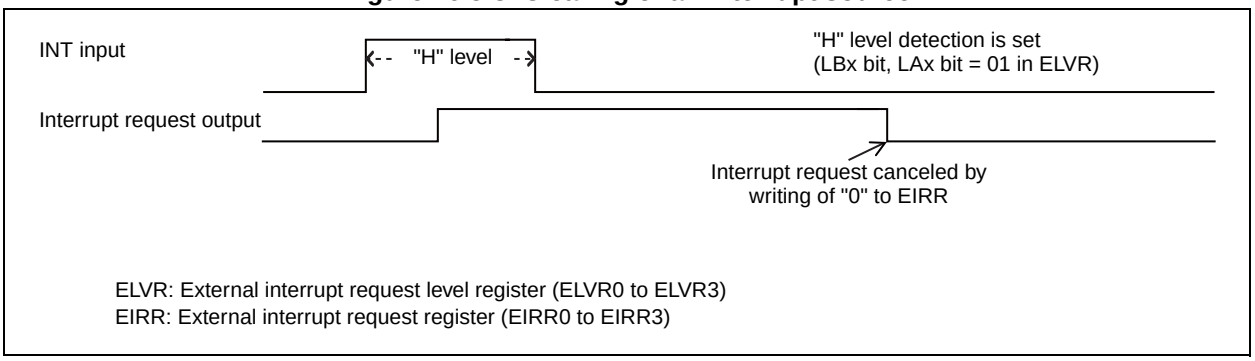

Input of external interrupt request
Edge/Level detection
External interrupt request register (EIRR0 to EIRR3)
Gate
Interrupt controller
Interrupt source maintained in same state even after cancellation of external interrupt request

Figure 16.5-3 Clearing of an interrupt source


INT input
"H" level
"H" level detection is set
(LBx bit, LAx bit = 01 in ELVR)
Interrupt request output
Interrupt request canceled by writing of "0" to EIRR
ELVR: External interrupt request level register (ELVR0 to ELVR3)
EIRR: External interrupt request register (EIRR0 to EIRR3)

16.5.2 Return from Standby Mode

■ Overview

External interrupt requests can be used for a return from standby mode (watch mode or stop mode). A signal already input to any of the INT0 to INT31 pins in standby mode in asynchronous input can be used for a return from standby mode.

■ Settings

Before a transition to standby mode, the following setting for the INT0 to INT31 pins must be made with the enable interrupt request registers (ENIR0 to ENIR3):

- Pins used for the return from standby mode: Enable interrupt request output.
- Pins not used for the return from standby mode: Disable interrupt request output.

■ Return operation

The MB91640/645 returns from standby mode when the effective level is detected in a signal input to the INT0 to INT31 pins in standby mode.

Table 16.5-1 shows the relationship between external interrupt request detection conditions and the levels for returning from standby mode.

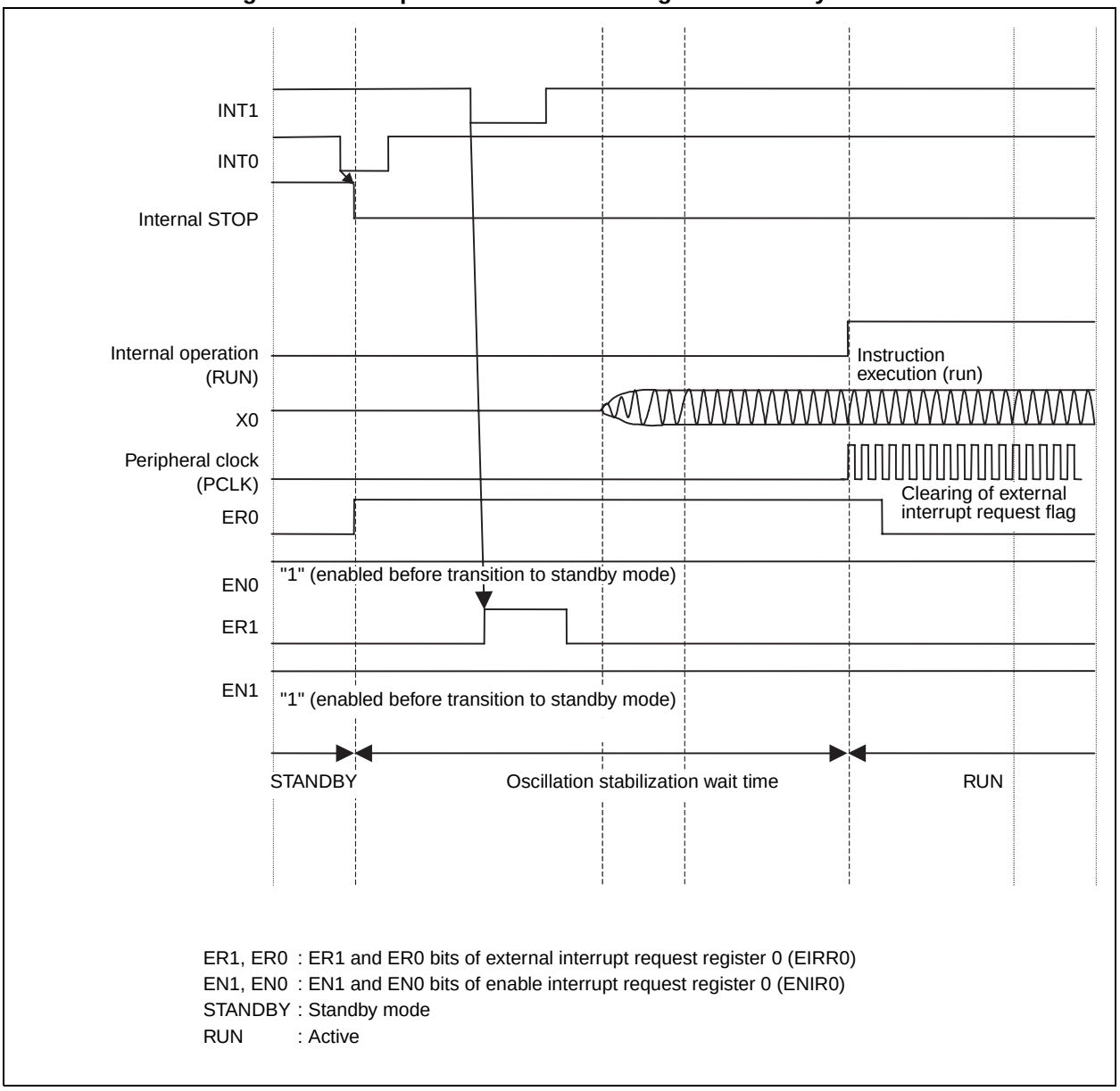
Table 16.5-1 Relationship between external interrupt request detection conditions and the levels for returning from standby mode

Detection Condition	LB31 to LB0	LA31 to LA0	Level for Returning from Standby Mode
"L" level detection	0	0	"L" level detection
"H" level detection	0	1	"H" level detection
Rising edge detection	1	0	"L" level detection at the INT0 to INT7 pins, and rising edge detection at the INT8 to INT31 pins
Falling edge detection	1	1	"H" level detection at the INT0 to INT7 pins, and falling edge detection at the INT8 to INT31 pins

After the MB91640/645 returns from standby mode, other external interrupt requests cannot be recognized until the oscillation stabilization wait time has elapsed. To output an external interrupt request after the MB91640/645 returns from standby mode, input an external interrupt request signal after the oscillation stabilization wait time has elapsed.

Figure 16.5-4 shows an example of operation at the time of return from standby mode, where the INT0 and INT1 pins are used.

Figure 16.5-4 Operation when returning from standby mode



ER1, ER0 : ER1 and ER0 bits of external interrupt request register 0 (EIRR0)
EN1, EN0 : EN1 and EN0 bits of enable interrupt request register 0 (ENIR0)
STANDBY : Standby mode
RUN : Active

16.5.3 Return from Sleep Mode

■ Overview

External interrupt requests can be used for a return from sleep mode.

■ Settings

Before a transition to sleep mode, the following setting for the INT0 to INT31 pins must be made with the enable interrupt request registers (ENIR0 to ENIR3):

- Pins used for the return from sleep mode: Enable interrupt request output.
- Pins not used for the return from sleep mode: Disable interrupt request output.

■ Return operation

The MB91640/645 returns from sleep mode when a signal at the specified level/edge is input to the INT0 to INT31 pins in sleep mode.

CHAPTER 17 Watchdog Timer

This chapter explains the functions and operations of the watchdog timer.

17.1 Overview

17.2 Configuration

17.3 Registers

17.4 Explanation of Operations and Setting Procedure Examples

17.1 Overview

The watchdog timer is a monitoring timer used to determine whether software hangs up or performs other abnormal operations.

■ Overview

If the watchdog timer is not cleared before the specified period has elapsed, it judges that software has hung up and outputs a reset request to the CPU. This reset request is called a watchdog reset request.

The operation of the watchdog timer requires that it be continually and periodically cleared before the specified period has elapsed. If an abnormal operation of software such as hanging up prevents it from being periodically cleared, it overflows and outputs a watchdog reset request.

- The watchdog timer counts cycles while a program is active on the CPU, and it stops counting while the CPU is stopped (in sleep mode, stop mode, or watch mode).
- The watchdog timer can detect a transition to standby mode (watch mode/stop mode), and it can output a watchdog reset request to the CPU.
- If an incorrect value is written to watchdog timer clear pattern register 0 (WDTCPR0), the watchdog timer outputs a watch reset request to the CPU.
- The following period can be selected as the watchdog timer period: peripheral clock (PCLK) x (2^9 to 2^{24})

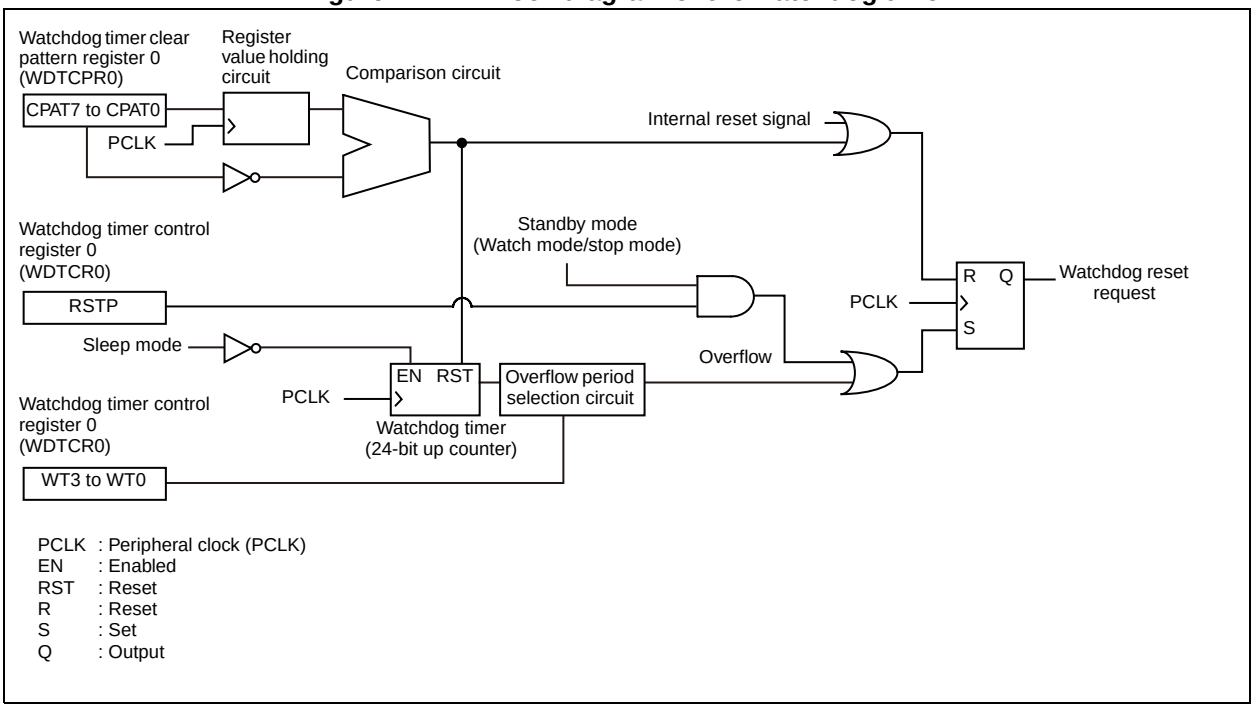
17.2 Configuration

This section shows the configuration of the watchdog timer.

■ Block diagram of the watchdog timer

Figure 17.2-1 is a block diagram of the watchdog timer.

Figure 17.2-1 Block diagram of the watchdog timer



- Watchdog timer control register 0 (WDTCR0)

  This register controls the operation of the watchdog timer.
- Watchdog timer clear pattern register 0 (WDTCPR0)

  This register activates and clears the watchdog timer.
- Watchdog timer

  This is a 24-bit up counter.
- Register value holding circuit

  This circuit retains the value written in watchdog timer clear pattern register 0 (WDTCPR0).
- Comparison circuit

  This circuit compares the value written in watchdog timer clear pattern register 0 (WDTCPR0) with the previous value that was written.
- Overflow period selection circuit

  This circuit selects the overflow period of the watchdog timer.

■ Clocks

Table 17.2-1 lists the clock used by the watchdog timer.

Table 17.2-1 Clock used by the watchdog timer

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

17.3 Registers

This section explains the configuration and functions of registers for the watchdog timer.

■ List of registers for the watchdog timer

Table 17.3-1 lists the registers for the watchdog timer.

Table 17.3-1 Registers for the watchdog timer

Abbreviated Register Name	Register Name	Reference
WDTCR0	Watchdog timer control register 0	17.3.1
WDTCPR0	Watchdog timer clear pattern register 0	17.3.2

17.3.1 Watchdog Timer Control Register 0 (WDTCR0)

This register controls the operation of the watchdog timer.

Figure 17.3-1 shows the bit configuration of watchdog timer control register 0 (WDTCR0).

Figure 17.3-1 Bit configuration of watchdog timer control register 0 (WDTCR0)

bit	7	6	5	4	3	2	1	0
	Reserved	RSTP	Reserved	Reserved	WT3	WT2	WT1	WT0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

This register can be written only prior to activation of the watchdog timer.

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6]: RSTP (Stop mode detection reset enable bit)

This bit specifies whether to enable output of a watchdog reset request at the transition time of the CPU to standby mode (watch mode/stop mode) while the watchdog timer is active.

Written Value	Explanation
0	Disables output of a watchdog reset request. The counting of the watchdog timer is suspended when a transition to standby mode (watch mode/stop mode) is detected, and it remains suspended until a return from standby mode.
1	Enables output of a watchdog reset request. A watchdog reset request is output when a transition to standby mode (watch mode/stop mode) is detected.

<Notes>

- To use standby mode (watch mode/stop mode), set "0" in this bit.
- This register can be written only before the watchdog timer is activated. If "1" is set in this bit after the watchdog timer is activated, standby mode (watch mode/stop mode) is detected and a watchdog reset request is output. Therefore, standby mode becomes unusable.

[bit5, bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit3 to bit0]: WT3 to WT0 (Watchdog timer period selection bits)

These bits select one of the following periods as the period from watchdog timer clearing to watchdog reset request output.

WT3 to WT0	Watchdog Timer Period
0000	PCLK x 2^{9}
0001	PCLK x 2^{10}
0010	PCLK x 2^{11}
0011	PCLK x 2^{12}
0100	PCLK x 2^{13}
0101	PCLK x 2^{14}
0110	PCLK x 2^{15}
0111	PCLK x 2^{16}
1000	PCLK x 2^{17}
1001	PCLK x 2^{18}
1010	PCLK x 2^{19}
1011	PCLK x 2^{20}
1100	PCLK x 2^{21}
1101	PCLK x 2^{22}
1110	PCLK x 2^{23}
1111	PCLK x 2^{24}

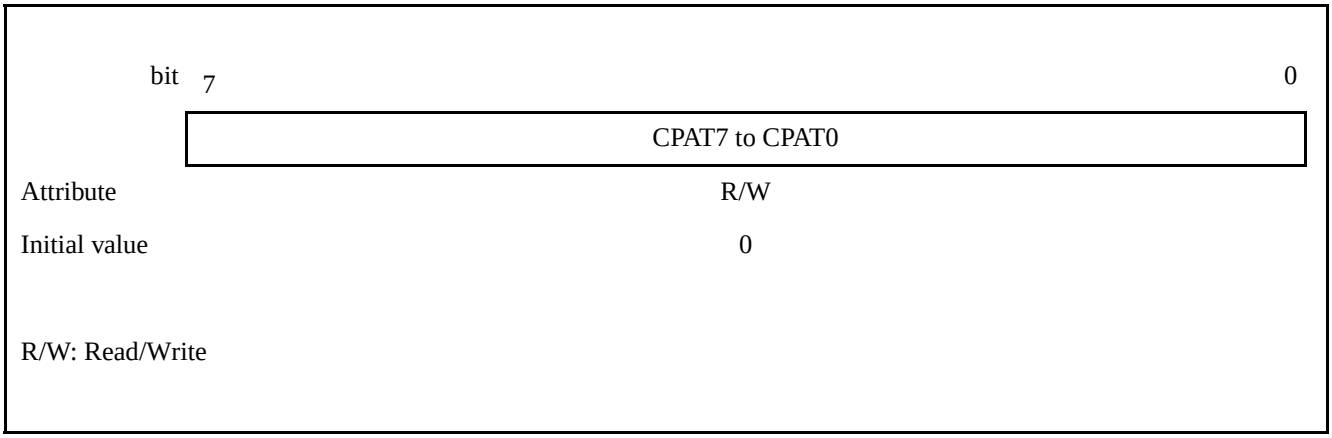
PCLK : Period of Peripheral clock (PCLK)

17.3.2 Watchdog Timer Clear Pattern Register 0 (WDTCPR0)

This register activates and clears the watchdog timer.

Figure 17.3-2 shows the bit configuration of watchdog timer clear pattern register 0 (WDTCPR0).

Figure 17.3-2 Bit configuration of watchdog timer clear pattern register 0 (WDTCPR0)



[bit7 to bit0]: CPAT7 to CPAT0 bits

The watchdog timer is activated when any value is written to this register after this device is reset.

To prevent a watchdog reset request from being output after the watchdog timer is activated, the timer must be cleared before the timer period has elapsed.

To clear the watchdog timer, invert the bit pattern written in these bits and write the inverted value to the bits.

For details of clearing the watchdog timer, see "■ Clearing the watchdog timer" in "17.4.1 Operations of the Watchdog Timer".

CPAT7 to CPAT0	In Case of Writing	In Case of Reading
Value obtained by inverting the written value	After being activated, the watchdog timer is cleared.	"0" is read.
Value other than that obtained by inverting the written value	A watchdog reset request is output immediately.	

17.4 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the watchdog timer. Also, examples of procedures for setting operating states are shown.

17.4.1 Operations of the Watchdog Timer

If the watchdog timer is not periodically cleared even though the program is designed to do so, a malfunction is judged to have occurred and the watchdog timer outputs a watchdog reset request to the CPU.

■ Overview

While the watchdog timer is operating, if it is not cleared before the specified period has elapsed, it judges that software has hung up and outputs a watchdog reset request to the CPU.

A watchdog reset request is also output if an incorrect value is written to watchdog timer clear pattern register 0 (WDTCPR0) or at the transition time of the CPU to standby mode (watch mode/stop mode).

Also, the watchdog timer stops the counting operation when the CPU is stopped.

■ Settings

To use the watchdog timer, specify the following with watchdog timer control register 0 (WDTCR0) before activating the watchdog timer:

- Period from watchdog timer clearing to the watchdog reset request output (WT3 to WT0 bits)
- Whether to enable output of a watchdog reset request at the transition time of the CPU to standby mode (watch mode/stop mode) (RSTP)

<Notes>

- The watchdog timer performs counting only while the CPU is operating. Therefore, the WT3 to WT0 bits must be set based on the setting of the number of program steps and the clock division setting.
- To use standby mode (watch mode/stop mode), set "0" in the RSTP bit.
- If "1" is set in the RSTP bit after the watchdog timer is activated, standby mode (watch mode/stop mode) cannot be used.

■ Operations

The watchdog timer is activated when any value is written to the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) after this device is reset. The counter value changes in sync with the rising edge of the peripheral clock (PCLK) while the CPU is active.

Unless the watchdog timer is cleared before the period specified by the WT3 to WT0 bits of watchdog timer control register 0 (WDTCR0) has elapsed, a watchdog reset request is output to the CPU.

Also, the watchdog timer temporarily stops counting while the CPU is stopped, such as during doze mode or sleep mode.

The value of the watchdog timer is not cleared while the counting is temporarily stopped. When the counting resumes, it starts from the value at which it was stopped.

<Notes>

- Even during DMA transfer with the DMA controller (DMAC), the watchdog timer continues counting as long as the CPU is operating.
- Since the peripheral clock (PCLK) is stopped during the oscillation stabilization wait time of the CPU source clock (SRCCLK), the watchdog timer also stops counting during this time.
- Sampling of the CPU operation state is performed using the peripheral clock (PCLK). Therefore, a change in the operating state that does not last longer than the period of the peripheral clock (PCLK) may be ignored.

■ Clearing the watchdog timer

The watchdog timer can be cleared by inverting the value written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) at the watchdog timer activation time and writing the inverted value to these bits.

For example, if "55_H" is written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) at the watchdog timer activation time, the watchdog timer can be cleared by writing the inverted value "AA_H" to the bits.

Clearing of the watchdog timer can be subsequently repeated by alternately writing "55_H" and "AA_H" to the CPAT7 to CPAT0 bits.

However, a watchdog reset request is output to the CPU when any value other than the inverted values is written to the CPAT7 to CPAT0 bits.

<Note>

If it is difficult to maintain the value written in these bits, writing of a value to them can be followed by writing of its inverted value (e.g., writing "AA_H" then writing "55_H") every time the watchdog timer is cleared.

■ Output of a watchdog reset request

The watchdog timer outputs a watchdog reset request to the CPU in any of the following cases:

- The period specified by the WT3 to WT0 bits of watchdog timer control register 0 (WDTCR0) has elapsed (overflow).
- The value written in the CPAT7 to CPAT0 bits of watchdog timer clear pattern register 0 (WDTCPR0) is different from the value obtained by inverting the written value.
- There is a transition by the CPU to standby mode (watch mode/stop mode) (a watchdog reset request may be output depending on the setting of the RSTP bit of watchdog timer control register 0 (WDTCR0)).

For details of the operations after output of a watchdog reset request, see "9.5 Explanation of Operations" of "CHAPTER 9 Reset".

CHAPTER 18 Watch Counter

This chapter explains the functions and operations of the watch counter.

18.1 Overview
18.2 Configuration
18.3 Registers
18.4 Interrupts
18.5 Explanation of Operations and Setting Procedure Examples
18.6 Notes on Use

18.1 Overview

The watch counter is a timer that counts down starting from the specified value, and it generates an interrupt request at the time that the 6-bit down counter enters an underflow condition.
Interrupt requests can be generated at a period ranging from 125 ms to 64 s.
This series has 1 built-in channel for the watch counter.
* This function is not available when the sub clock (SBCLK) is not being used.

■ Overview

- The count clock can be selected from 4 types of clock, and interrupt requests can be set to be generated at an interval ranging from a minimum of 125 ms to a maximum of 64 s.

  Table 18.1-1 lists the count clocks and counting periods.

Table 18.1-1 Count clocks and counting periods

Period of Count Clock	Counting Period (F_{CL} = 32.768 kHz)
$2^{12}/F_{CL}$	125 ms
$2^{13}/F_{CL}$	250 ms
$2^{14}/F_{CL}$	500 ms
$2^{15}/F_{CL}$	1 s

F_{CL} Sub clock (SBCLK) frequency

- A number between 0 and 63 can be set as the value used for counting by the 6-bit down counter.

  If "60" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 1 minute. If "0" is the count value used for a counting period of 1 second, an interrupt request is generated at an interval of 64 seconds.
- An interrupt request can be generated at the time that the 6-bit down counter enters an underflow condition.

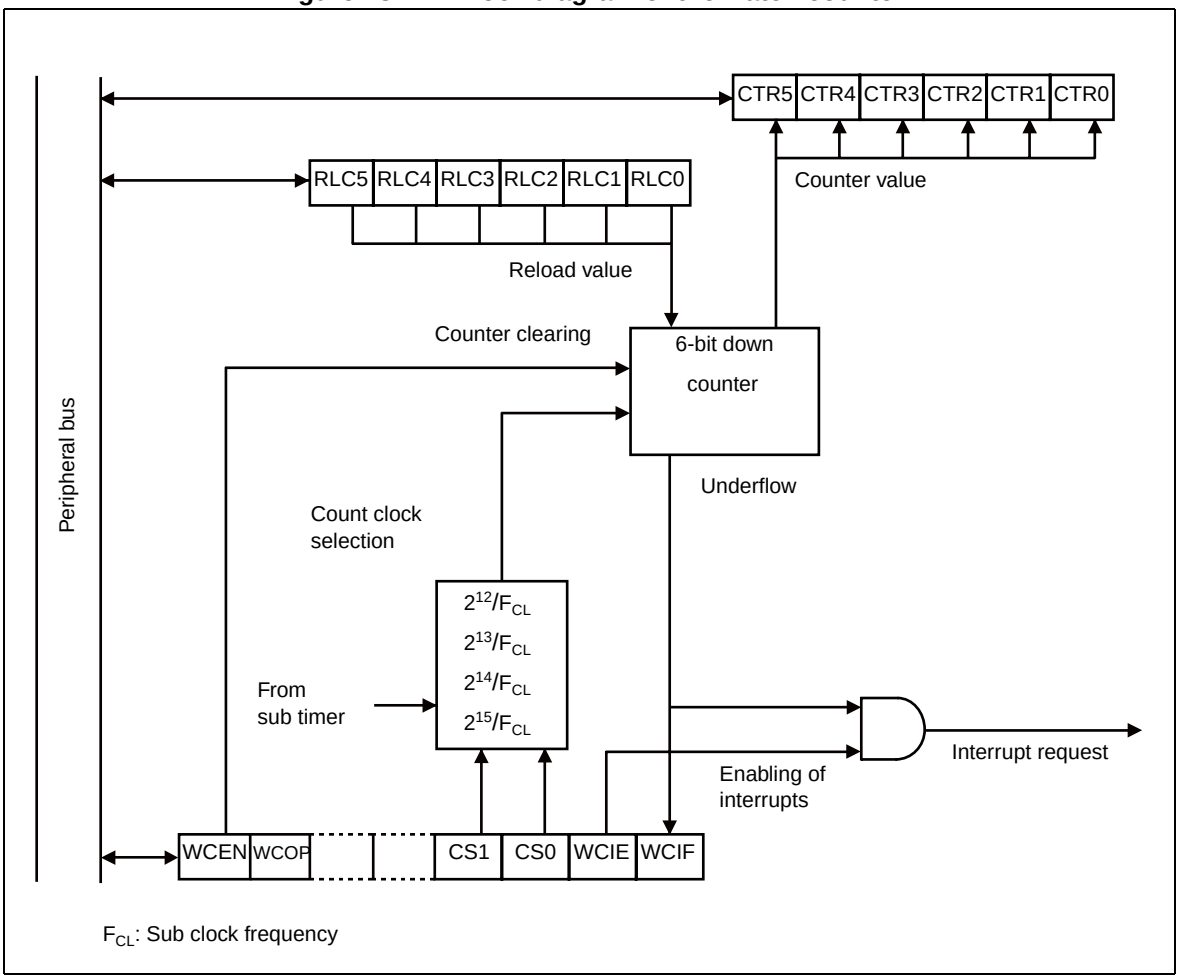
18.2 Configuration

This section shows the watch counter configuration.

■ Block diagram of the watch counter

Figure 18.2-1 is a block diagram of the watch counter.

Figure 18.2-1 Block diagram of the watch counter



- 6-bit down counter

  This is the 6-bit down counter of the watch counter. It reloads the value set in the watch counter reload register (WCRL) and starts a countdown.

- Watch counter reload register (WCRL)

  This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in this register.

- Watch counter read register (WCRD)

  This register reads the value in the 6-bit down counter. Also, the register can be read to check the count value.

- Watch counter control register (WCCR)

  This register controls the operation of the watch counter.

■ Clocks

Table 18.2-1 lists the clocks used by the watch counter.

Table 18.2-1 Clocks used by the watch counter

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Sub timer output	Sub timer period*

* The sub timer period is specified by the STS2 to STS0 bits in the sub timer control register (STMCR). For details of the sub timer, see "CHAPTER 7 Sub Timer".

18.3 Registers

This section explains the configurations and functions of the registers for the watch counter.

■ List of registers for the watch counter

Table 18.3-1 lists the registers for the watch counter.

Table 18.3-1 Registers for the watch counter

Abbreviated Register Name	Register Name	Reference
WCRL	Watch counter reload register	18.3.1
WCCR	Watch counter control register	18.3.2
WCRD	Watch counter read register	18.3.3

18.3.1 Watch Counter Reload Register (WCRL)

This register specifies the value used by the watch counter to start counting. The 6-bit down counter counts down starting from the value set in the register.

The register specifies the reload value for the 6-bit down counter. If the 6-bit down counter enters an underflow condition, the value in this register is reloaded in the 6-bit down counter, and the countdown is restarted.

Figure 18.3-1 shows the bit configuration of the watch counter reload register (WCRL).

Figure 18.3-1 Bit configuration of the watch counter reload register (WCRL)

bit	7	6	5	4	3	2	1	0
	Unde-fined	Unde-fined	RLC5	RLC4	RLC3	RLC2	RLC1	RLC0
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit5 to bit0]: RLC5 to RLC0 (Counter reload value setting bits)

These bits set the reload value for the 6-bit down counter.

The 6-bit down counter counts downwards from the reload value and enters an underflow condition when its value reaches "1". If "0" is set in these bits, it performs 64 countdowns from "63" to "0".

<Notes>

- If the value of these bits is changed to another value while the 6-bit down counter is active, an underflow occurs and the new value is then reloaded.
- If the value of these bits is changed to another value at the same time that an underflow interrupt request is generated, the correct value is not reloaded. Be sure to rewrite the value of these bits either when the watch counter is stopped or in the interrupt processing routine before an interrupt request is generated.
- To verify whether the reload value is correctly set, read this register.

18.3.2 Watch Counter Control Register (WCCR)

This register selects a count clock for the watch counter or enables/disables generation of interrupt requests. The register also enables/disables the operation of the watch counter.

Figure 18.3-2 shows the bit configuration of the watch counter control register (WCCR).

Figure 18.3-2 Bit configuration of the watch counter control register (WCCR)

bit	7	6	5	4	3	2	1	0
	WCEN	WCOP	Unde-fined	Unde-fined	CS1	CS0	WCIE	WCIF
Attribute	R/W	R	-	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

[bit7]: WCEN (Watch counter operation enable bit)

This bit enables/disables the operation of the watch counter.

Written Value	Explanation
0	The watch counter is disabled/stopped. The value in the 6-bit down counter is cleared to "000000_B".
1	The watch counter is enabled/started.

<Notes>

- Output of the sub timer is used for the count clock of the watch counter, and the peripheral clock (PCLK) is used for the settings of each register. Since the sub timer and peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: Count clock period) may occur at the count start time, depending on the time at which "1" is written to this bit.
- Before writing "1" to this bit to start the operation of the watch counter, verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0).

[bit6]: WCOP (Watch counter operating state flag bit)

This bit indicates the operating state of the watch counter.

Read Value	Explanation
0	The watch counter is stopped.
1	The watch counter is active.

[bit5, bit4]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit3, bit2]: CS1, CS0 (Count clock selection bits)

These bits set the count clock of the watch counter.

CS1	CS0	Count Clock
0	0	$2^{12}/F_{CL}$
0	1	$2^{13}/F_{CL}$
1	0	$2^{14}/F_{CL}$
1	1	$2^{15}/F_{CL}$

F_{CL} Sub clock (SBCLK) frequency

<Note>

The following conditions must be satisfied when the information in these bits is changed:

- WCEN bit = 0 (watch counter operation disabled)
- WCOP bit = 0 (watch counter stopped)

[bit1]: WCIE (Interrupt request enable bit)

This bit specifies whether to generate an underflow interrupt request at the time that the 6-bit down counter enters an underflow condition (WCIF bit = 1).

Written Value	Explanation
0	Disables generation of an underflow interrupt request.
1	Enables generation of an underflow interrupt request.

[bit0]: WCIF (Interrupt request flag bit)

This bit indicates whether the 6-bit down counter has entered an underflow condition.

If "1" is set in the WCIE bit, an interrupt request is generated when "1" is set in this bit.

WCIF	In Case of Reading	In Case of Writing
0	The down counter has not entered an underflow condition.	This bit is cleared to "0".
1	The down counter has entered an underflow condition.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

18.3.3 Watch Counter Read Register (WCRD)

This register reads the value in the 6-bit down counter.

Figure 18.3-3 shows the bit configuration of the watch counter read register (WCRD).

Figure 18.3-3 Bit configuration of the watch counter read register (WCRD)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	CTR5	CTR4	CTR3	CTR2	CTR1	CTR0
Attribute	-	-	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

-: Undefined

<Note>

If the 6-bit down counter is operating when its value is read, the register value must be read twice and verified to be the same value.

18.4 Interrupts

The 6-bit down counter enters an underflow condition when the value in the 6-bit down counter becomes "000001_B", and an underflow interrupt request is then generated.

Table 18.4-1 outlines the interrupts that can be used with the watch counter.

Table 18.4-1 Interrupts of the watch counter

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Underflow interrupt request	WCIF=1 for WCCR	WCIE=1 for WCCR	Write "0" to the WCIF bit for WCCR

WCCR: watch counter control register (WCCR)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling generation of interrupt requests.

  - Clear interrupt requests before enabling the generation of interrupt requests.
  - Clear interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- To set the interrupt level corresponding to the interrupt vector number, use an interrupt control register (ICR00 to ICR47). For details of setting interrupt levels, see "CHAPTER 10 Interrupt Controller".

18.5 Explanation of Operations and Setting Procedure Examples

This section explains operations of the watch counter. Also, examples of procedures for setting the operating state are shown.

18.5.1 Operations of the Watch Counter

The watch counter is a timer that counts down starting from the value set in the watch counter reload register (WCRL), and it generates an interrupt request at the time that the 6-bit down counter enters an underflow condition.

To operate the watch counter, follow the procedure below.

1. Select a count clock by using the CS1 and CS0 bits of the watch counter control register (WCCR).
2. Set a count value to the RLC5 to RLC0 bits in the watch counter reload register (WCRL).
3. Enable the operation of the watch counter by using the WCEN bit (WCEN = 1) of the watch counter control register (WCCR).

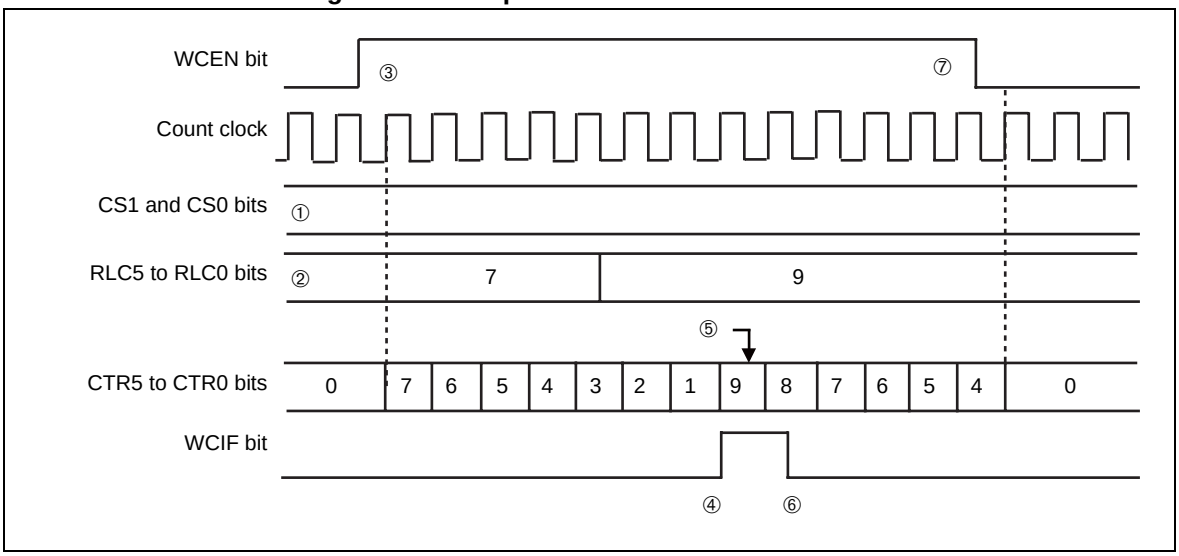
   Start a countdown. Counting is performed at the rising edge of the count clock.
4. If the 6 -bit down counter enters an underflow condition, the value of the WCIF bit in the watch counter control register (WCCR) is changed to "1".

   At this time, if generation of underflow interrupt requests has been enabled by the WCIE bit in the watch counter control register (WCCR), an underflow interrupt request is generated.

   Also, the value that is set in the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is reloaded in the 6-bit down counter, and the countdown is restarted.
5. If the value of the RLC5 to RLC0 bits in the watch counter reload register (WCRL) is changed to another value while the watch counter is active, the watch counter is updated with the new value at the next reload time.
6. The underflow interrupt request is cleared when "0" is written to the WCIF bit in the watch counter control register (WCCR).
7. The 6-bit down counter is cleared to "000000_B" and the counting operation is stopped when "0" is written to the WCEN bit in the watch counter control register (WCCR).

Figure 18.5-1 shows the operation of the watch counter.

Figure 18.5-1 Operation of the watch counter



<Notes>

- Output of the sub timer is used for the count clock of the watch counter, and the peripheral clock (PCLK) is used for the settings of each register. Since the sub timer and peripheral clock (PCLK) are not synchronized, an error of up to 1T (T: Count clock period) may occur at the count start time, depending on the time at which "1" is written to the WCEN bit in the watch counter control register (WCCR).
- Since the count clock from the sub timer is also stopped when the sub clock (SBCLK) is stopped, the 6-bit down counter is stopped too. Even when the sub clock (SBCLK) starts operating again, the watch counter cannot count counter values correctly. Before using the watch counter when the sub clock (SBCLK) starts operating again, be sure to write "0" to the WCEN bit in the watch counter control register (WCCR) to clear the counter value to "000000_B".
- Even when the CPU is operating in watch mode, the watch counter continues operating as long as the sub timer is operating. The watch mode of the CPU can be canceled with the watch counter interrupt processing routine.
- If the sub timer is cleared while the watch counter is active, counting values correctly may become impossible. Stop the watch counter by using the WCEN bit (WCEN = 0) of the watch counter control register (WCCR), and then clear the sub timer.
- After the watch counter is stopped by writing "0" to the WCEN in the watch counter control register (WCCR), be sure to verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0) in the watch counter control register (WCCR) before reactivating the watch counter by using the WCEN bit (WCEN = 1).

18.6 Notes on Use

Note the following points about using the watch counter.

■ Notes on operations

- If the sub timer is cleared while the watch counter is active, counting values correctly may become impossible. Stop the watch counter by using the WCEN bit (WCEN = 0) of the watch counter control register (WCCR), and then clear the sub timer.
- After the watch counter is stopped by the WCEN bit (WCEN = 0) in the watch counter control register (WCCR), be sure to verify that the watch counter is stopped by checking the WCOP bit (WCOP = 0) in the watch counter control register (WCCR) before reactivating the watch counter by using the WCEN bit (WCEN = 1).
- Since the watch counter uses output of the sub timer as the count clock, the setting of the sub timer must not be changed while the watch counter is active.
- The watch counter enters an underflow condition when it counts downwards from "000001_B". It counts downwards from the reload value to "1". If the value is set to "0", it performs 64 countdowns.

CHAPTER 19 32-bit Free-Run Timer

This chapter explains the functions and operations of the 32-bit free-run timer.

19.1 Overview
19.2 Configuration
19.3 Pins
19.4 Registers
19.5 Interrupts
19.6 An Explanation of Operations and Setting Procedure Examples

19.1 Overview

The 32-bit free-run timer is an up-counter that counts up to the predetermined value.
After counting up to the specified value, the free-run timer clears the value and starts counting again or generates an interrupt request. The count value is also used as the reference time for 32-bit output compare or 32-bit input capture.
This series microcontroller has 2 built-in channels for the 32-bit free-run timer.

■ Overview

The 32-bit free-run timer is part of the compare timer. The compare timer comprises the following three peripheral functions:

- 32-bit free-run timer (2 channels)
- 32-bit output compare (8 channels)

  See "CHAPTER 21 32-bit Output Compare".
- 32-bit input capture (8 channels)

  See "CHAPTER 20 32-bit Input Capture".

This chapter explains the 32-bit free-run timer.

- Count clock: One of the following can be selected:
  - Internal clock (peripheral clock)

    Can be selected from 9 types, which are peripheral clocks (PCLK) divided by 1, 2, 4, 8, 16, 32, 64,128, and 256.
  - External clock
- Interrupt request: Can be issued in the following cases:

  The count value of the 32-bit free-run timer matches the preset value (compare clear interrupt).
- Of the values of the 2 channels of the 32-bit free-run timer, one can be selected for use as the reference time for 32-bit output compare and 32-bit input capture.

19.2 Configuration

The 32-bit free-run time is part of the compare timer. The following is a block diagram of the compare timer and the 32-bit free-run timer.

■ Compare timer block diagram

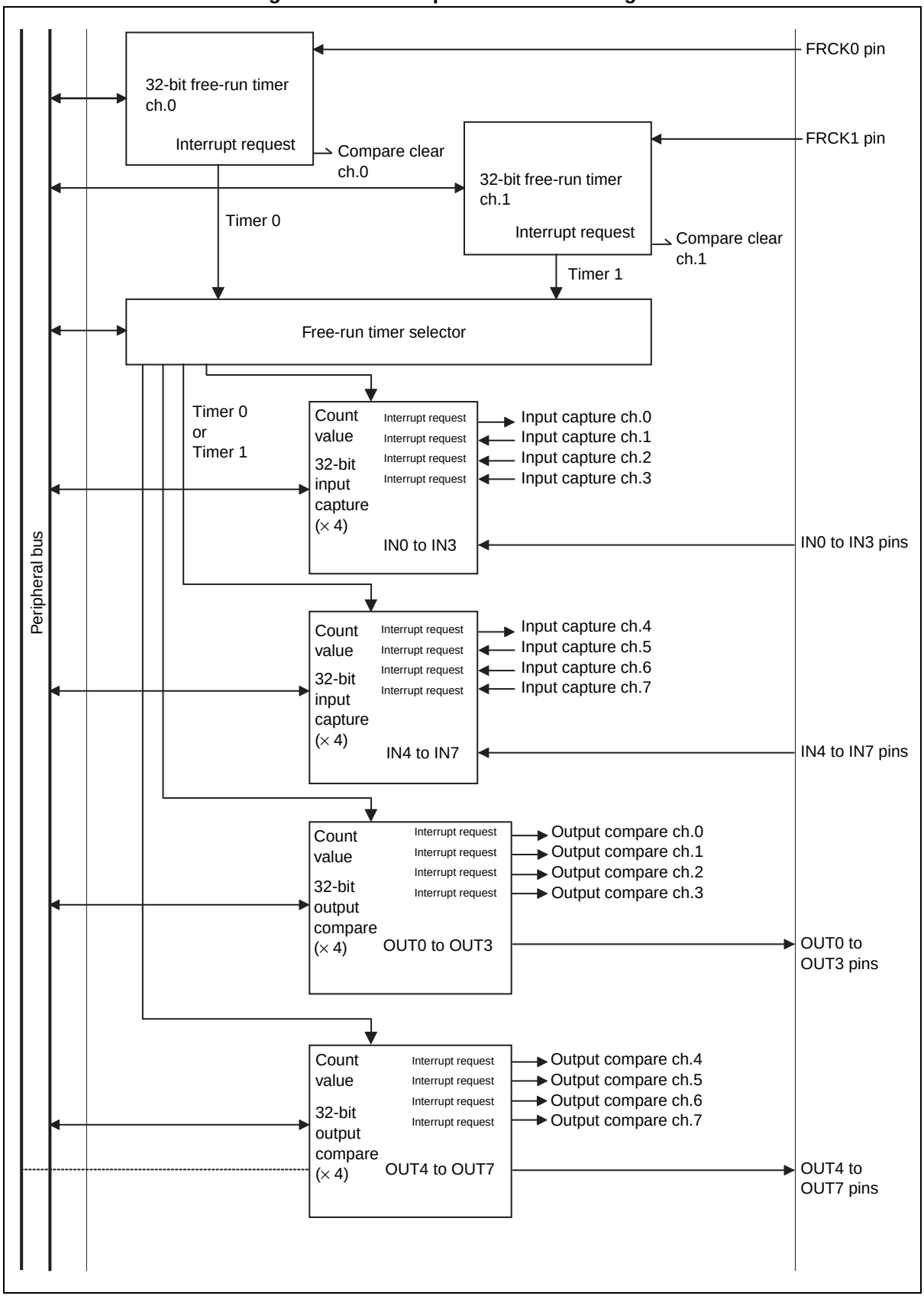
The compare timer consists of the following blocks.

- 32-bit free-run timer
- Free-run timer selector

  The free-run timer selector selects the 32-bit free-run timer used as the reference time for the 32-bit output compare and 32-bit input capture.
- 32-bit input capture (8 channels)
- 32-bit output compare (8 channels)

Figure 19.2-1 is a compare timer block diagram.

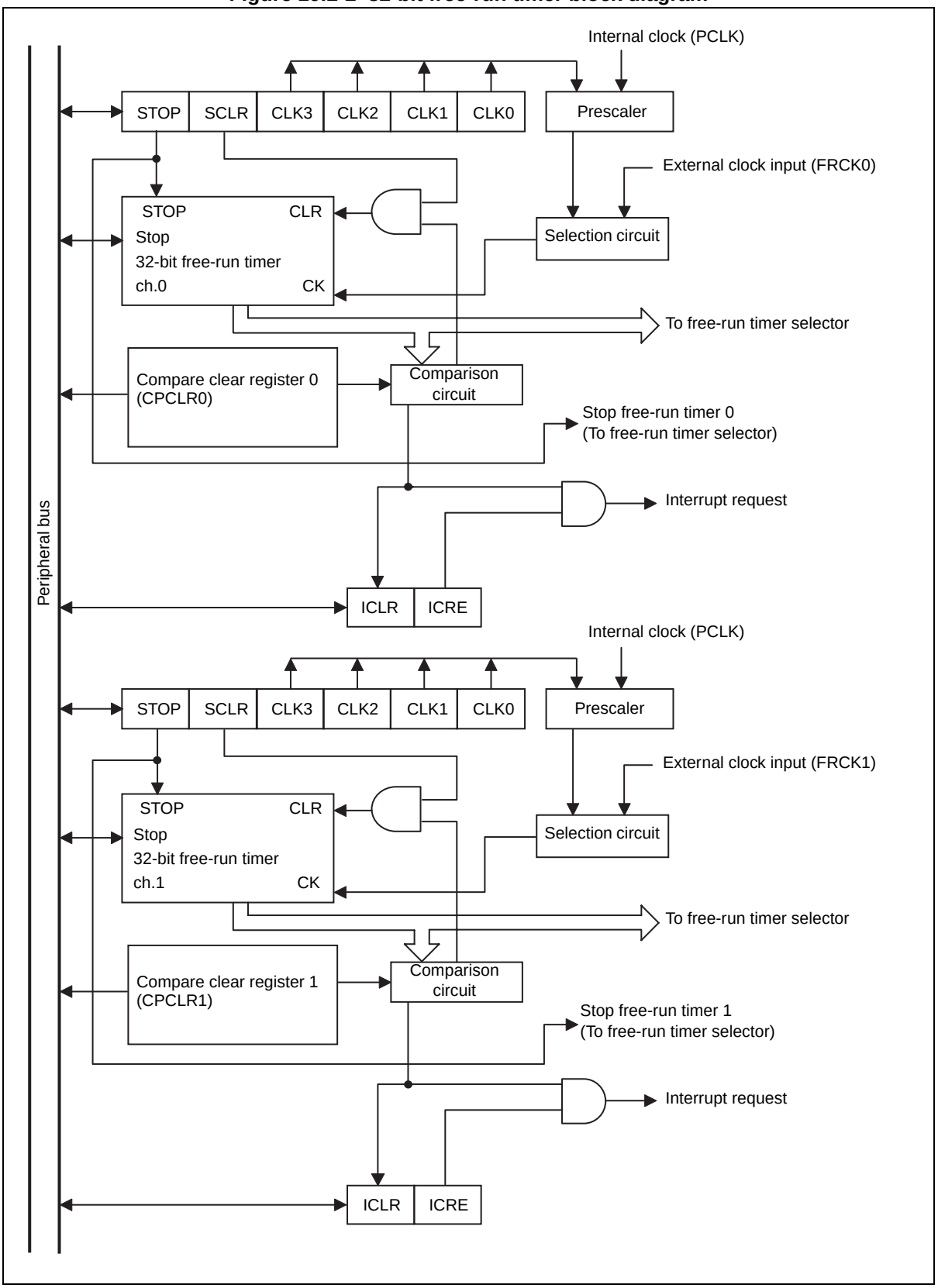
Figure 19.2-1 Compare timer block diagram



■ 32-bit free-run timer block diagram

Figure 19.2-2 is a block diagram of the 32-bit free-run timer.

Figure 19.2-2 32-bit free-run timer block diagram



- 32-bit free-run timer

  This counter counts up to the value that is set in the compare clear register (CPCLR0, CPCLR1)
- Timer status control register upper/lower (TCCSH0/TCCSL0, TCCSH1/TCCSL1)

  This register controls the operation of the 32-bit free-run timer.
- Compare clear register (CPCLR0, CPCLR1)

  The 32-bit up counter counts up to the value that is set in this register.
- Timer data register (TCDT0, TCDT1)

  This register is used to set the value with which the timer starts counting or to read the current count value.
- Prescaler

  When the internal clock (peripheral clock) is selected for the count clock, the prescaler divides the peripheral clock (PCLK)
- Selection circuit

  The selection circuit selects whether to use the internal clock (peripheral clock) or external clock (FRCK0, FRCK1) for the count clock.
- Comparison circuit

  The comparison circuit compares the count value of the 32-bit free-run timer and the value set in the compare clear register (CPCLR0, CPCLR1).

■ Clocks

Table 19.2-1 lists the clocks used for the 32-bit free-run timer.

Table 19.2-1 Clocks used for 32-bit free-run timer

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Created through division of the peripheral clock (PCLK).
	External clock	Input from the FRCK0 and FRCK1 pins

19.3 Pins

This section explains the pins used by the 32-bit free-run timer.

■ Overview

- FRCK0 and FRCK1 pins

  These pins are 32-bit free-run timer external clock input pins. These pins are multiplexed pins.

  To use these pins as the FRCK0 and FRCK1 pins of the 32-bit free-run timer, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 19.3-1 shows the relationship between channels and pins.

Table 19.3-1 Relationship between channels and pins

Channel	Input Pin
0	FRCK0
1	FRCK1

19.4 Registers

This section explains the configuration and functions of the registers used by the 32-bit free-run timer.

■ 32-bit free-run timer registers

Table 19.4-1 lists the registers of the 32-bit free-run timer.

Table 19.4-1 32-bit free-run timer registers

Channel	Abbreviated Register Name	Register Name	Reference
Common	FRTSEL	Free-run timer select register	19.4.1
0	CPCLR0	Compare clear register 0	19.4.2
	TCCSH0/TCCSL0	Timer status control register upper0/lower0	19.4.4
	TCDT0	Timer data register 0	19.4.3
1	CPCLR1	Compare clear register 1	19.4.2
	TCCSH1/TCCSL1	Timer status control register upper1/lower1	19.4.4
	TCDT1	Timer data register 1	19.4.3

19.4.1 Free-Run Timer Select Register (FRTSEL)

This register specifies the channel for use as the reference time for 32-bit output compare and 32-bit input capture, of the 2 channels of 32-bit free-run timer.

Figure 19.4-1 shows the bit configuration of the free-run timer select register (FRTSEL).

Figure 19.4-1 Bit configuration of free-run timer select register (FRTSEL)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	FRS1	FRS0
Attribute	-	-	-	-	-	-	R/W	R/W
Initial value	X	X	X	X	X	X	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7 to bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1, bit0]: FRS1, FRS0 (free-run timer selection bit)

These bits select the 32-bit free-run timer channel used as the reference time for the 32-bit output compare and 32-bit input capture.

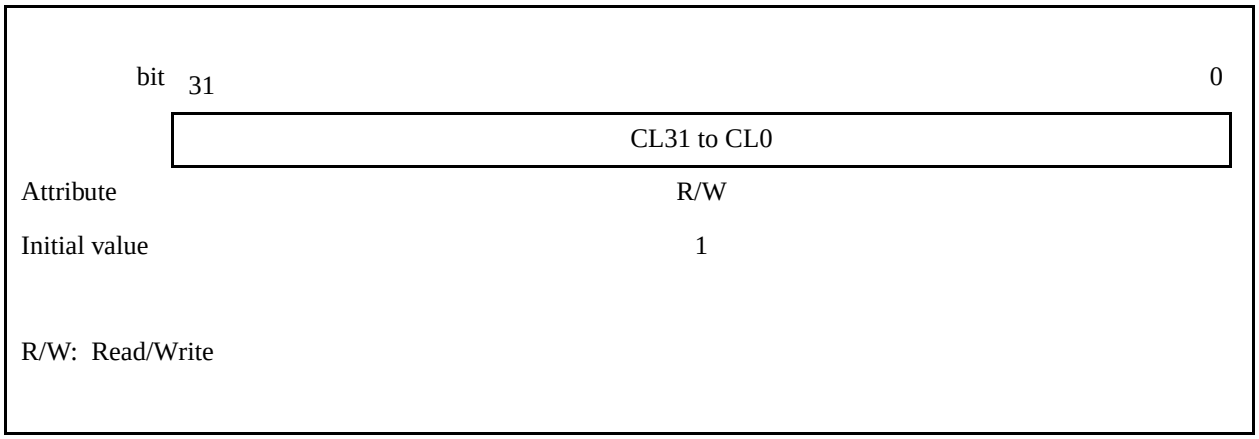
FRS1	FRS0	Explanation	
		Free-run Timer Channel	Use
0	0	ch.0	32-bit output compare (ch.0 to ch.7) 32-bit input capture (ch.0 to ch.7)
0	1	ch.0	32-bit output compare (ch.0 to ch.3) 32-bit input capture (ch.0 to ch.3)
		ch.1	32-bit output compare (ch.4 to ch.7) 32-bit input capture (ch.4 to ch.7)
1	0	ch.0	32-bit output compare (ch.0 to ch.7)
		ch.1	32-bit input capture (ch.0 to ch.7)
1	1	Setting prohibited	

19.4.2 Compare Clear Register (CPCLR0, CPCLR1)

This register sets the comparison value of the 32-bit free-run timer.
When the 32-bit free-run timer counts up and reaches the value that is set in this register, the count value of the 32-bit free-run timer is cleared to "0000 0000_H".

Figure 19.4-2 shows the bit configuration of the compare clear register (CPCLR0, CPCLR1).

Figure 19.4-2 Bit configuration of compare clear register (CPCLR0, CPCLR1)



<Notes>

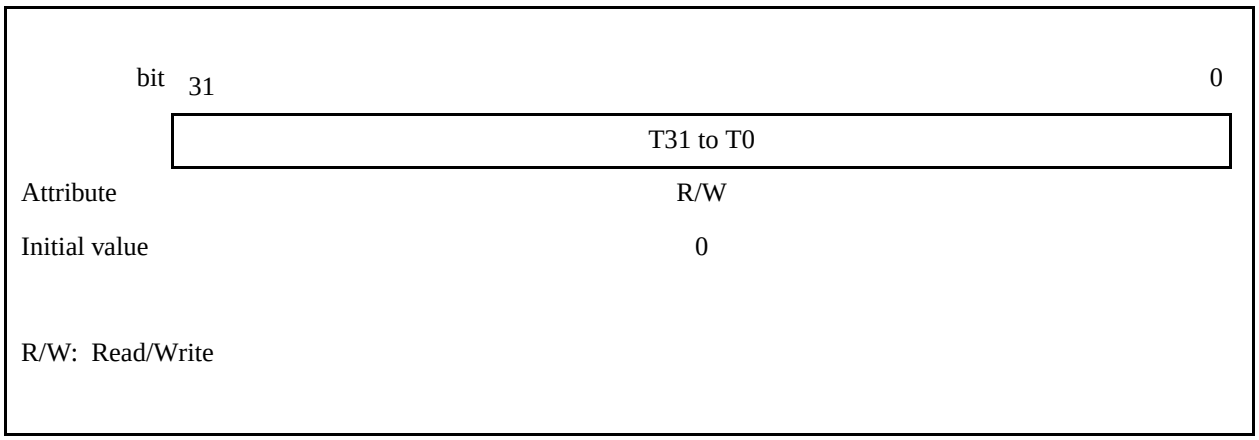
- Rewrite this register while the 32-bit free-run timer is stopped.
  The 32-bit free-run timer is stopped when the STOP bit of the timer status control register lower (TCCSL0, TCCSL1) is "1".
- Be sure to access this register in units of words.

19.4.3 Timer Data Register (TCDT0, TCDT1)

This register is used to set the value with which the 32-bit free-run timer starts counting or to read the current count value.

Figure 19.4-3 shows the bit configuration of the timer data register (TCDT0, TCDT1).

Figure 19.4-3 Bit configuration of timer data register (TCDT0, TCDT1)



The 32-bit free-run timer counts up starting from the value written to this register. If this register is read, the count value of the 32-bit free-run timer is read.

<Notes>

- Rewrite this register while the 32-bit free-run timer is stopped.
  The 32-bit free-run timer is stopped when the STOP bit of the timer status control register lower (TCCSL0, TCCSL1) is "1".
- Be sure to access this register in units of half word.
- The write value and read value of this register are different.
- If one of the following occurs, the count value of the 32-bit free-run timer (the value of this register) is promptly cleared to "0000 0000_H".
  - This device is reset.
  - "1" is written to the SCLR bit of the timer status control register lower (TCCSL0, TCCSL1).
  - The count value of the 32-bit free-run time matches the value of the compare clear register (CPCLR0, CPCLR1).

19.4.4 Timer Status Control Register Upper/Lower (TCCSH0/TCCSL0, TCCSH1/TCCSL1)

This register controls the operation of the 32-bit free-run timer.

Figure 19.4-4 shows the bit configuration of the timer status control register upper/lower (TCCSH0/ TCCSL0, TCCSH1/TCCSL1).

Figure 19.4-4 Bit configuration of timer status control register upper/lower (TCCSH0/TCCSL0, TCCSH1/TCCSL1)

Timer status control register upper (TCCSH0 TCCSH1)

bit	15	14	13	12	11	10	9	8
	ECKE	Undefined	Undefined	Undefined	Undefined	Undefined	ICLR	ICRE
Attribute	R/W	-	-	-	-	-	R/W	R/W
Initial value	0	X	X	X	X	X	0	0

Timer status control register lower (TCCSL0, TCCSL1)

bit	7	6	5	4	3	2	1	0
	Undefined	STOP	Undefined	SCLR	CLK3	CLK2	CLK1	CLK0
Attribute	-	R/W	-	R/W	R/W	R/W	R/W	R/W
Initial value	X	1	X	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15]: ECKE (Clock selection bit)

This bit selects the count clock of the 32-bit free-run timer.

Written Value	Explanation
0	Selects the internal clock (peripheral clock).
1	Selects an external clock.

An internal clock (peripheral clock) is generated by dividing the peripheral clock (PCLK). If an internal clock (peripheral clock) is selected, CLK3 to CLK0 bits must be used to select the division rate of the peripheral clock (PCLK).

An external clock is input through the FRCK0 and FRCK1 pins. When an external clock is selected, the timer counts on both edges of the signal input through the FRCK0 or FRCK1 pin.

<Notes>

- The count clock changes as soon as this bit is changed.
- Rewrite this bit while the 32-bit free-run timer, 32-bit input capture, and 32-bit output compare are all stopped.

[bit14 to bit10]: Reserved bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9]: ICLR (compare clear interrupt request flag bit)

This bit indicates that the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1).

If "1" is set in the ICRE bit when this bit is "1", a compare clear interrupt request is generated.

ICLR	In Case of Reading	In Case of Writing
0	The count value does not match the preset value.	This bit is cleared to "0".
1	The count value matches the preset value.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit8]: ICRE (compare clear interrupt request enable bit)

This bit is specifies whether to generate a compare clear interrupt request when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1) (ICLR bit = 1).

Written Value	Explanation
0	Disables generation of compare clear interrupt requests.
1	Enables generation of compare clear interrupt requests.

[bit7]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit6]: STOP (timer operation enable bit)

This bit enables (starts) or disables (stops) the count operation of the 32-bit free-run timer.

Written Value	Explanation
0	Enables (starts) the count function.
1	Disables (stops) the count function.

<Note>

When the 32-bit free-run timer is stopped, the 32-bit output compare is also stopped.

[bit5]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit4]: SCLR (timer clear bit)

This bit clears the count value of the 32-bit free-run timer to "0000 0000_H".

SCLR	In Case of Writing	In Case of Reading
0	Does not clear the count value.	"0" is read.
1	Clears the count value.	

<Note>

When this bit is set to "1", the count value is cleared at the next count clock timing.

[bit3 to bit0]: CLK3 to CLK0 (clock frequency selection bits)

These bits select the division rate of the peripheral clock (PCLK) when the internal clock (peripheral clock) is selected for the count clock of the 32-bit free-run timer,

The count cycle is determined by using the division rate selected by these bits and the peripheral clock (PCLK) frequency.

Table 19.4-2 provides an example of count cycles that are set according to the relationship between the values written to these bits and the peripheral clock (PCLK).

Table 19.4-2 Example of written values and count cycles

CLK3	CLK2	CLK1	CLK0	PCLK Division Rate	PCLK Frequency				
					32 MHz	16 MHz	8 MHz	4 MHz	1 MHz
0	0	0	0	Divided by 1	31.25 ns	62.5 ns	125 ns	0.25 µs	1 µs
0	0	0	1	Divided by 2	62.5 ns	125 ns	0.25 µs	0.5 µs	2 µs
0	0	1	0	Divided by 4	125 ns	0.25 µs	0.5 µs	1 µs	4 µs
0	0	1	1	Divided by 8	0.25 µs	0.5 µs	1 µs	2 µs	8 µs
0	1	0	0	Divided by 16	0.5 µs	1 µs	2 µs	4 µs	16 µs
0	1	0	1	Divided by 32	1 µs	2 µs	4 µs	8 µs	32 µs
0	1	1	0	Divided by 64	2 µs	4 µs	8 µs	16 µs	64 µs
0	1	1	1	Divided by 128	4 µs	8 µs	16 µs	32 µs	128 µs
1	0	0	0	Divided by 256	8 µs	16 µs	32 µs	64 µs	256 µs

PCLK: Peripheral clock (PCLK)

<Notes>

- Do not use any settings other than those listed in Table 19.4-2.
- The count clock changes as soon as this bit is rewritten.
- Rewrite this bit while the 32-bit free-run timer, 32-bit input capture, and 32-bit output compare are all stopped.

19.5 Interrupts

An interrupt request (compare clear interrupt request) is generated when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1).

Table 19.5-1 outlines the interrupts that can be used with the 32-bit free-run timer.

Table 19.5-1 Interrupts of the 32-bit free-run timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Compare clear interrupt request	ICLR=1 for TCCSH	ICRE=1 for TCCSH	Write "0" to the ICLR bit for TCCSH

TCCSH: timer status control register upper (TCCSH0, TCCSH1)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use an interrupt control register (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

19.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the 32-bit free-run timer. Also, examples of procedures for setting the operating state are shown.

■ Overview

The 32-bit free-run timer uses an internal clock (peripheral clock) or an external clock as count clock and counts up starting from the value set in the timer data register (TCDT0, TCDT1) to the value set in the compare clear register (CPCLR0, CPCLR1).

- Internal clock (peripheral clock)

  Can be selected from 9 types, which are peripheral clocks (PCLK) divided by 1, 2, 4, 8, 16, 32, 64,128, and 256.

- External clock

  The timer counts up at both edges. The count start timing varies depending on the initial value of the external clock input through the FRCK0 or FRCK1 pin.

The count value of the 32-bit free-run timer is used as the reference time for 32-bit output compare or 32-bit input capture.

■ Timer clearing

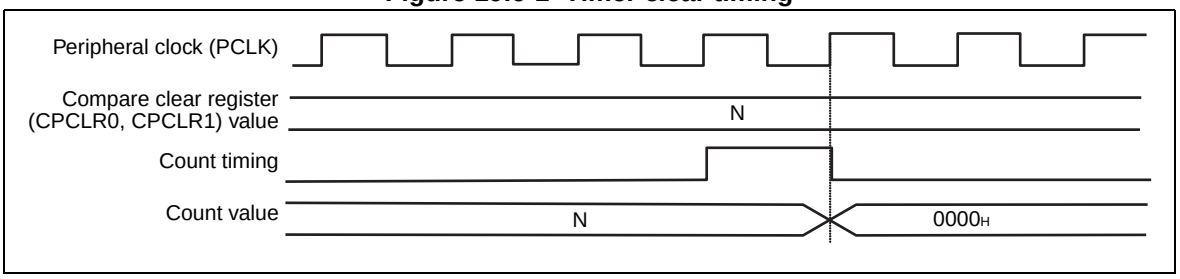
The count value of the 32-bit free-run timer is promptly cleared when one of the following conditions is met:

- This count value matches the value that is set in the compare clear register (CPCLR0, CPCLR1).
- The SCLR bit of the timer status control register lower (TCCSL0, TCCSL1) is set to 1 to clear the count value of the 32-bit free-run timer.
- "0000 0000_H" is written to the timer data register (TCDT0, TCDT1) while the 32-bit free-run timer is stopped.
- This device is reset.

When the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1), the count value is cleared in synchronization with the count timing.

Figure 19.6-1 shows the timer clear timing.

Figure 19.6-1 Timer clear timing



19.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected

A divided peripheral clock (PCLK) is used as the count clock.

■ Count operation

When the STOP bit of the timer status control register lower (TCCSL0, TCCSL1) is set to 0 to enable the 32-bit free-run timer, the timer counts up starting from the value set in the timer data register (TCDT0, TCDT1) to the value set in the compare clear register (CPCLR0, CPCLR1).

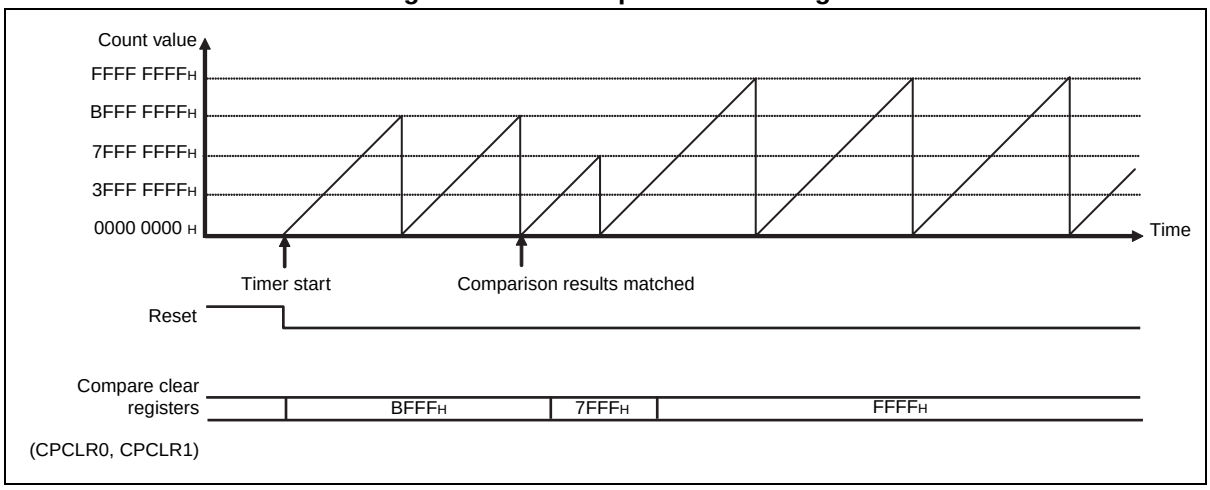
■ Compare clear

When the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1), the count value is cleared in synchronization with the count timing (compare clear).

After compare clear, the timer starts counting again.

Figure 19.6-2 shows the compare clear timing.

Figure 19.6-2 Compare clear timing



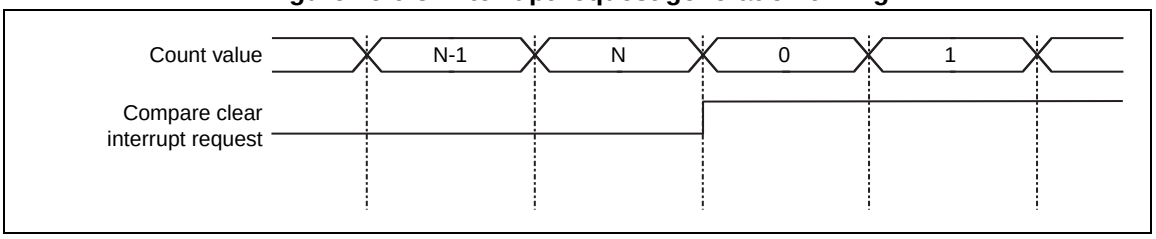
■ Interrupt processing

An interrupt request can be generated when the count value of the 32-bit free-run timer matches the value set in the compare clear register (CPCLR0, CPCLR1).

The interrupt request can be cleared by writing "0" to the ICLR bit of the timer status control register upper (TCCSH0 TCCSH1).

Figure 19.6-3 shows the interrupt request generation timing.

Figure 19.6-3 Interrupt request generation timing



19.6.2 Operation When an External Clock Is Selected

The external clock input through the FRCK0 or FRCK1 pin is used as the count clock.

■ Count operation

Upon detection of a valid edge through the FRCK0 or FRCK1 pin while the STOP bit of the timer status control register lower (TCCSL0, TCCSL1) is set to 0 to enable the 32-bit free-run timer, the timer counts up starting from the value set in the timer data register (TCDT0, TCDT1) to the value set in the compare clear register (CPCLR0, CPCLR1).

The count timing varies depending on the signal level input through the FRCK0 or FRCK1 pin when the free-run timer is enabled.

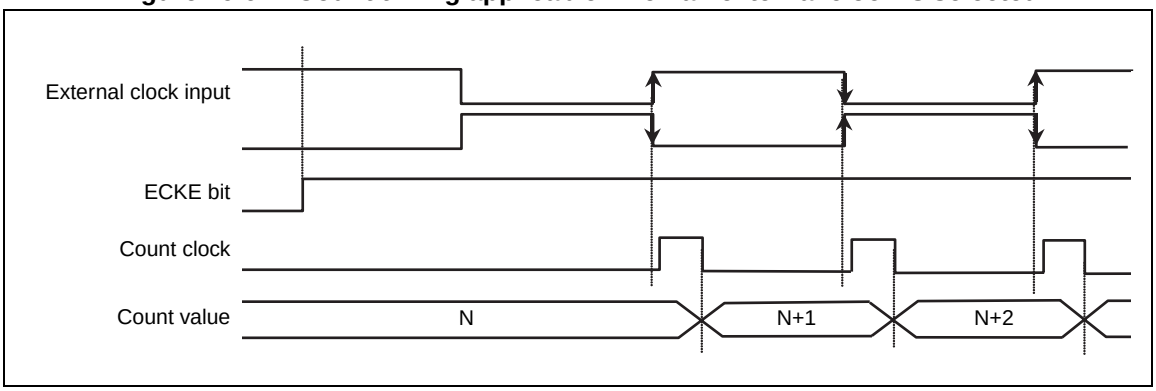
Table 19.6-1 lists the count timings applicable when an external clock is selected.

Table 19.6-1 Count timings applicable when an external clock is selected

Signal Level When Timer Is Enabled	Count Timing
"H" level	Starts counting at a rising edge and thereafter counts up at both edges.
"L" level	Starts counting at a falling edge and thereafter counts up at both edges.

Figure 19.6-4 shows the count timing applicable when an external clock is selected (ECKE=1).

Figure 19.6-4 Count timing applicable when an external clock is selected



■ Compare clear

Same as when an internal clock (peripheral clock) is selected. See "■ Compare clear" in "19.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected".

■ Interrupt processing

Same as when an internal clock (peripheral clock) is selected. See "■ Interrupt processing" in "19.6.1 Operation When an Internal Clock (Peripheral Clock) Is Selected".

CHAPTER 20 32-bit Input Capture

This chapter explains the functions and operations of the 32-bit input capture.

20.1 Overview
20.2 Configuration
20.3 Pins
20.4 Registers
20.5 Interrupts
20.6 An Explanation of Operations and Setting Procedure Examples

20.1 Overview

Upon detection of an input signal edge that is set in advance, the 32-bit input capture saves the value of the 32-bit free-run timer at the time.
This series microcontroller has 8 built-in input capture channels.

■ Overview

The 32-bit input capture is part of the compare timer. The compare timer comprises the following three functions:

- 32-bit free-run timer (2 channels)

  See "CHAPTER 19 32-bit Free-Run Timer".
- 32-bit output compare (8 channels)

  See "CHAPTER 21 32-bit Output Compare".
- 32-bit input capture (8 channels)

This chapter explains the 32-bit input capture.

- One of the following three triggers can be selected to save the value of the 32-bit free-run timer.
  - Rising edge
  - Falling edge
  - Both edges
- An interrupt request can be generated upon detection of an input signal edge that is set in advance.
- Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is saved by the 32-bit input capture can be selected.

  For details of the procedure for selecting the 32-bit free-run timer, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

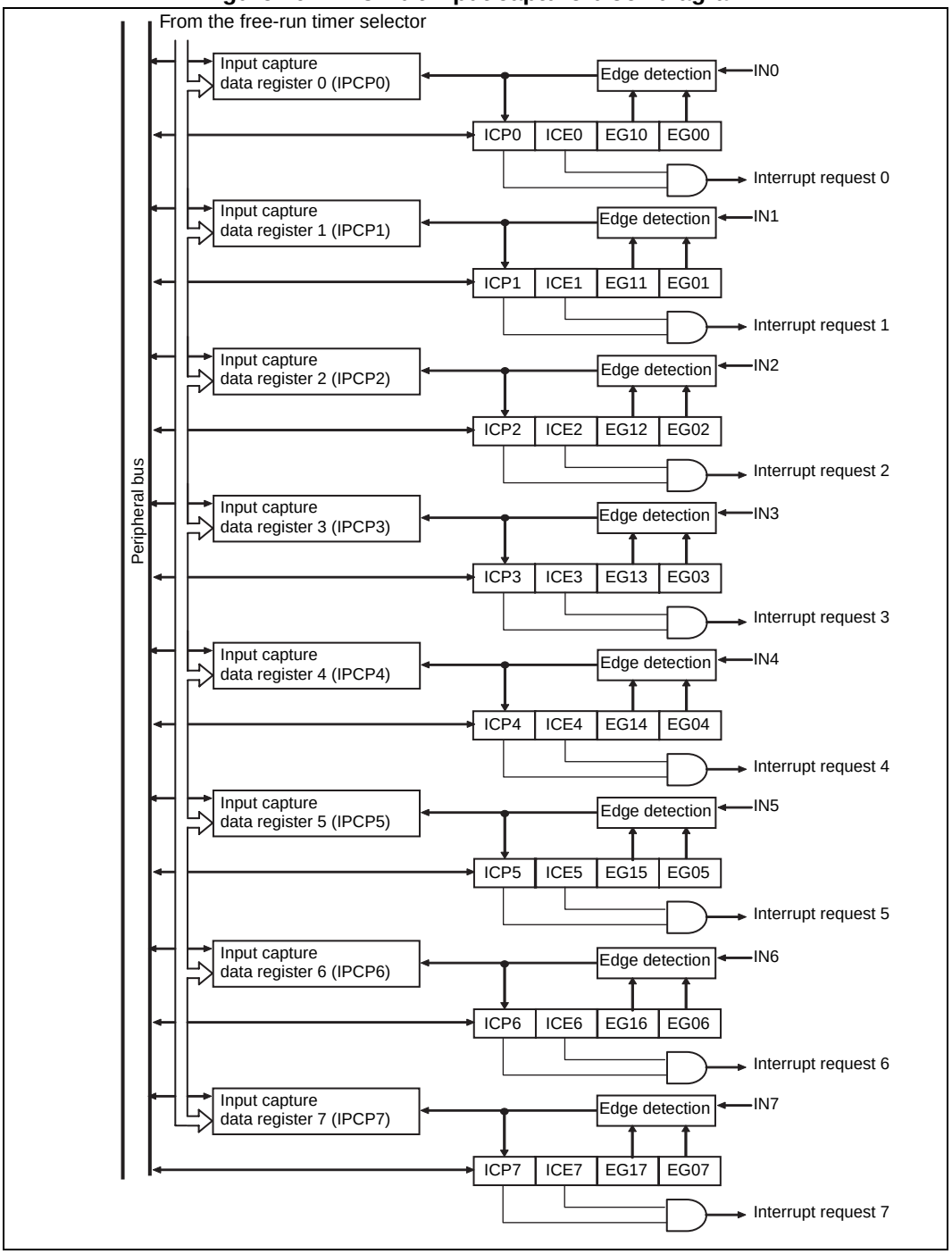
20.2 Configuration

This section explains the configuration of the 32-bit input capture.

■ 32-bit input capture block diagram

Figure 20.2-1 is a block diagram of the 32-bit input capture.

Figure 20.2-1 32-bit input capture block diagram



- Input capture data registers (IPCP0 to IPCP7)
  Free-run timer values are saved to these registers.
- Input capture status control registers (ICS01 to ICS67)
  These registers are used to control the operation and state of the 32-bit input capture.

<Note>

For details of the compare timer block diagram, see "■ Compare timer block diagram" in "CHAPTER 19 32-bit Free-Run Timer".

■ Clocks

Table 20.2-1 lists the clock used for the 32-bit input capture.

Table 20.2-1 Clock used for 32-bit input capture

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

20.3 Pins

This section explains the pins used by the 32-bit input capture.

■ Overview

- IN0 to IN7 pins

  Input pins of 32-bit input capture. These pins are multiplexed pins. To use these pins as input pins of the 32-bit input capture, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 20.3-1 lists the relationship between channels and pins.

Table 20.3-1 Relationship between channels and pins

Channel	Input Pin
0	IN0
1	IN1
2	IN2
3	IN3
4	IN4
5	IN5
6	IN6
7	IN7

20.4 Registers

This section explains the configuration and functions of registers used by the 32-bit input capture.

■ Registers of 32-bit input capture

Table 20.4-1 lists the registers of the 32-bit input capture.

Table 20.4-1 Registers of 32-bit input capture

Channel	Abbreviated Register Name	Register Name	Reference
Common	FRTSEL	Free-run timer select register	19.4.1
Common to 0 and 1	ICS01	Input capture status control register 01	20.4.1
Common to 2 and 3	ICS23	Input capture status control register 23	20.4.1
Common to 4 and 5	ICS45	Input capture status control register 45	20.4.1
Common to 6 and 7	ICS67	Input capture status control register 67	20.4.1
0	IPCP0	Input capture data register 0	20.4.2
1	IPCP1	Input capture data register 1	20.4.2
2	IPCP2	Input capture data register 2	20.4.2
3	IPCP3	Input capture data register 3	20.4.2
4	IPCP4	Input capture data register 4	20.4.2
5	IPCP5	Input capture data register 5	20.4.2
6	IPCP6	Input capture data register 6	20.4.2
7	IPCP7	Input capture data register 7	20.4.2

20.4.1 Input Capture Status Control Registers (ICS01 to ICS67)

These registers are used to control the operation and state of the 32-bit input capture.

Figure 20.4-1 shows the bit configuration of the input capture status control register (ICS01 to ICS67).

Figure 20.4-1 Bit configuration of input capture status control register (ICS01 to ICS67)

bit	7	6	5	4	3	2	1	0
	ICPm	ICPn	ICEm	ICEn	EG1m	EG0m	EG1n	EG0n
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7, bit6]: ICPm, ICPn (interrupt request flag bit)

Each of these bits indicates that a valid edge has been detected at pins IN0 to IN7. When this bit is "1" while ICEm or ICEn bit is set to "1", an edge detection interrupt request is generated.

The ICPm bit corresponds to the odd-numbered channel, and the ICPn bit corresponds to the even-numbered channel.

ICPm, ICPn	In Case of Reading	In Case of Writing
0	A valid edge is not detected.	This bit is cleared to "0".
1	A valid edge is detected.	Ignored

Table 20.4-2 lists the relationship between the ICPm bits and ICPn bits and channels.

Table 20.4-2 Relationship between bits and channels

Input Capture Status Registers	ICPm Bit	Supported Channel	ICPn Bit	Supported Channel
ICS01	ICP1	ch.1	ICP0	ch.0
ICS23	ICP3	ch.3	ICP2	ch.2
ICS45	ICP5	ch.5	ICP4	ch.4
ICS67	ICP7	ch.7	ICP6	ch.6

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit5, bit4]: ICEm, ICEn (interrupt request enable bits)

Each of these bits specifies whether to generate an edge detection interrupt request when a valid edge is detected through pins IN0 to IN7 (ICPm, ICPn=1).

The ICEm bit corresponds to the odd-numbered channel, and the ICEn bit corresponds to the even-numbered channel.

Written Value	Explanation
0	Disables generation of edge detection interrupt requests.
1	Enables generation of edge detection interrupt requests.

Table 20.4-3 shows the relationship between the ICEm bits and ICEn bits and channels.

Table 20.4-3 Relationship between bits and channels

Input Capture Status Registers	ICEm Bit	Supported Channel	ICEn Bit	Supported Channel
ICS01	ICE1	ch.1	ICE0	ch.0
ICS23	ICE3	ch.3	ICE2	ch.2
ICS45	ICE5	ch.5	ICE4	ch.4
ICS67	ICE7	ch.7	ICE6	ch.6

[bit3, bit2]: EG1m, EG0m (edge selection bits)

These bits select a valid edge for the 32-bit input capture of the odd-numbered channel.

When the edge selected here is detected, the value of the 32-bit free-run timer is saved to the input capture data register (IPCP0 to IPCP7).

EG1m	EG0m	Explanation
0	0	No edge detected (input capture stopped)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

Table 20.4-4 shows the relationship between the EG1m and EG0m bits and channels.

Table 20.4-4 Relationship between bits and channels

Input Capture Status Registers	EG1m, EG0m Bits	Supported Channel
ICS01	EG11, EG01	ch.1
ICS23	EG13, EG03	ch.3
ICS45	EG15, EG05	ch.5
ICS67	EG17, EG07	ch.7

<Note>

If a value other than "00" is written to these bits, the operation of the corresponding channel is enabled at the same time as a valid edge is selected.

[bit1, bit0]: EG1n, EG0n (edge selection bits)

These bits select a valid edge for the 32-bit input capture of the even-numbered channel.

When the edge selected here is detected, the value of the 32-bit free-run timer is saved to the input capture data register (IPCP0 to IPCP7).

EG1n	EG0n	Explanation
0	0	No edge detected (input capture stopped)
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

The bit names of EG1n and EG0n vary depending on the channel.

Table 20.4-5 shows the relationship between bits and channels.

Table 20.4-5 Relationship between bits and channels

Input Capture Status Registers	EG1n, EG0n Bits	Supported Channel
ICS01	EG10, EG00	ch.0
ICS23	EG12, EG02	ch.2
ICS45	EG14, EG04	ch.4
ICS67	EG16, EG06	ch.6

<Note>

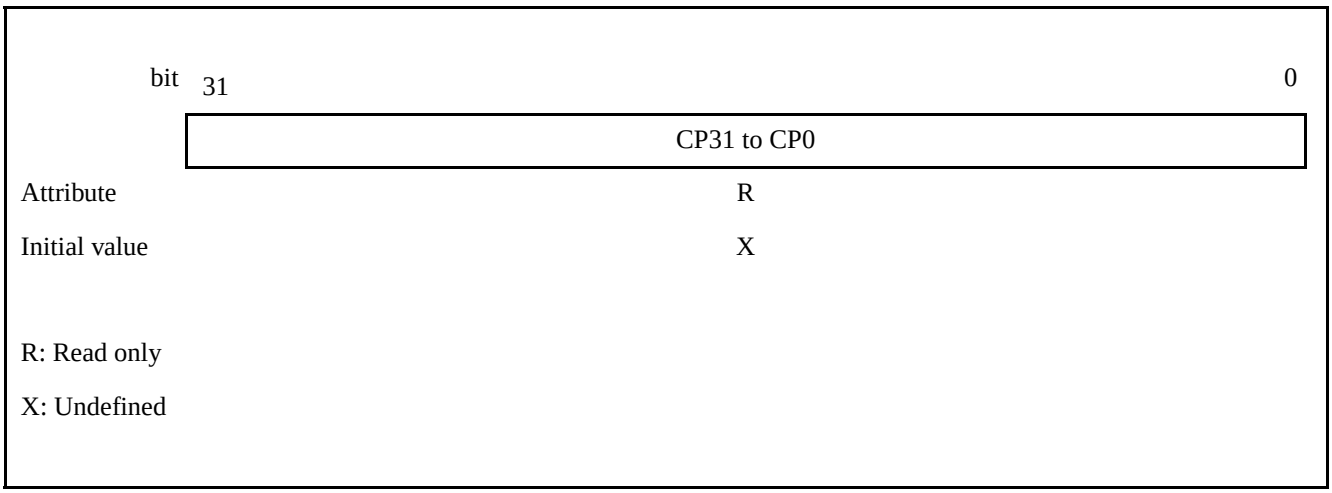
If a value other than "00" is written to these bits, the operation of the corresponding channel is enabled at the same time as a valid edge is selected.

20.4.2 Input Capture Data Register (IPCP0 to IPCP7)

This register saves the value of the 32-bit free-run timer. When a valid edge is detected in the input signal through pins IN0 to IN7, the value of the 32-bit free-run timer is saved to this register.

Figure 20.4-2 shows the bit configuration of the input capture data register (IPCP0 to IPCP7).

Figure 20.4-2 Bit configuration of input capture data register (IPCP0 to IPCP7)



<Notes>

- Be sure to read this register in units of words.
- Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is to be saved to this register varies depending on the free-run timer select register (FRTSEL) setting. For details, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

20.5 Interrupts

Upon detection of a valid edge in the input signal through pins IN0 to IN7, an interrupt request is generated (edge detection interrupt request).

Table 20.5-1 outlines the interrupts that can be used with the 32-bit input capture.

Table 20.5-1 Interrupts of the 32-bit input capture

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Edge detection interrupt request	Even-numbered channel: ICPn=1 for ICS Odd-numbered channel: ICPm=1 for ICS	Even-numbered channel: ICEn=1 for ICS Odd-numbered channel: ICEm=1 for ICS	Write "0" to the next bit. Even-numbered channel: ICPn bit for ICS Odd-numbered channel: ICPm bit for ICS

ICS: input capture status control register (ICS01 to ICS67)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests at the same time with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

20.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the 32-bit input capture. Also, examples of procedures for setting the operating state are shown.

20.6.1 Explanation of 32-bit Input Capture Operation

Upon detection of an input signal edge that is set in advance, the 32-bit input capture saves the value of the 32-bit free-run timer at the time.

■ Operation

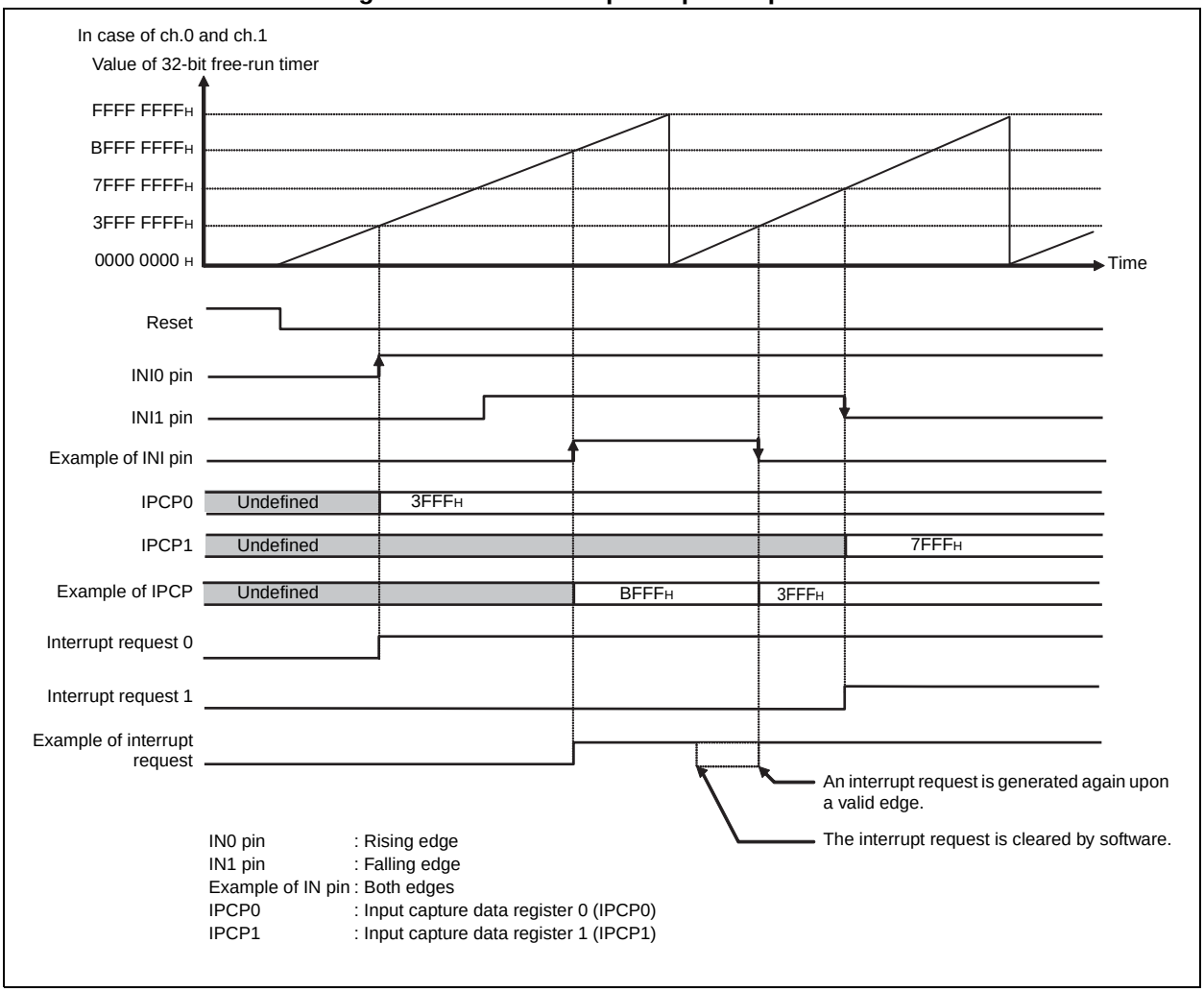
Selecting a valid edge with the following bits of the input capture status control register (ICS01 to ICS67) enables 32-bit input capture operation.

- Selecting valid edge of odd-numbered channel/enabling operation: EG1m, EG0m
- Selecting valid edge of even-numbered channel/enabling operation: EG1n, EG0n

When a valid edge is detected at pins IN0 to IN7 while 32-bit input capture operation is enabled, the value of the 32-bit free-run timer at the time is saved to the input capture data register (IPCP0 to IPCP7). If interrupt request generation has been enabled, an edge detection interrupt request is generated.

Figure 20.6-1 shows the 32-bit input capture operation.

Figure 20.6-1 32-bit input capture operation

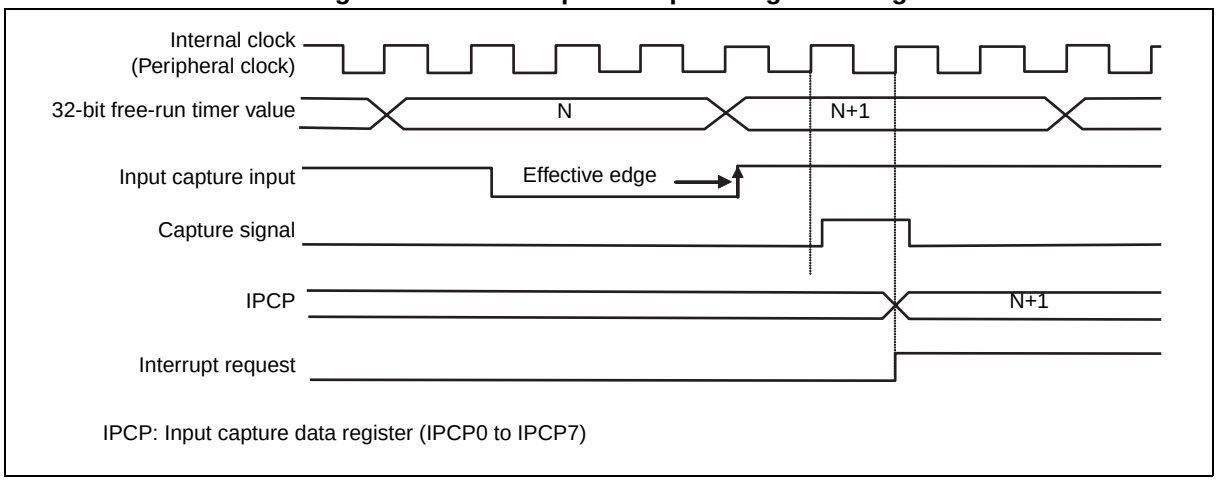


<Note>

Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is to be saved varies depending on the free-run timer select register (FRTSEL) setting. For details, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

When a valid edge is detected, a capture signal is generated to synchronize with the internal clock (peripheral clock). The generation of interrupt requests and the saving of 32-bit free-run timer values are performed based on the capture signals. Figure 20.6-2 shows an example of capture signal timing.

Figure 20.6-2 Example of capture signal timing



CHAPTER 21 32-bit Output Compare

This chapter explains the functions and operations of the 32-bit output compare.

21.1 Overview
21.2 Configuration
21.3 Pins
21.4 Registers
21.5 Interrupts
21.6 An Explanation of Operations and Setting Procedure Examples

21.1 Overview

After 32-bit free-run timer counts up to the preset value, the 32-bit output compare function inverts the level of output from a pin or generates an interrupt request.
This series microcontroller has 8 built-in channels for the 32-bit output compare.

■ Overview

The 32-bit output compare is part of the compare timer. The compare timer comprises the following three functions:

- 32-bit free-run timer (2 channels)

  See "CHAPTER 19 32-bit Free-Run Timer".
- 32-bit output compare (8 channels)
- 32-bit input capture (8 channels)

  See "CHAPTER 20 32-bit Input Capture".

This chapter explains the 32-bit output compare.

- 2 channels of the 32-bit output compare can be used either independently of each other or as a pair.

  If the 2 channels of the 32-bit output compare are used as a pair, comparison can be performed by 2 channels at one time and thus the CPU load can be reduced.

  The combinations of channels that can be used as pairs are as follows:

  - ch.0 and ch.1
  - ch.2 and ch.3
  - ch.4 and ch.5
  - ch.6 and ch.7
- The output levels at the OUT0 to OUT7 pins at the time of activation of the 32-bit output compare can be set.
- An interrupt request can be generated when the count value of the 32-bit free-run timer matches the preset value (compare value).
- Of the 2 channels of 32-bit free-run timer, the channel for use as the 32-bit output compare can be selected.

  For details of how to select the 32-bit free-run timer, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

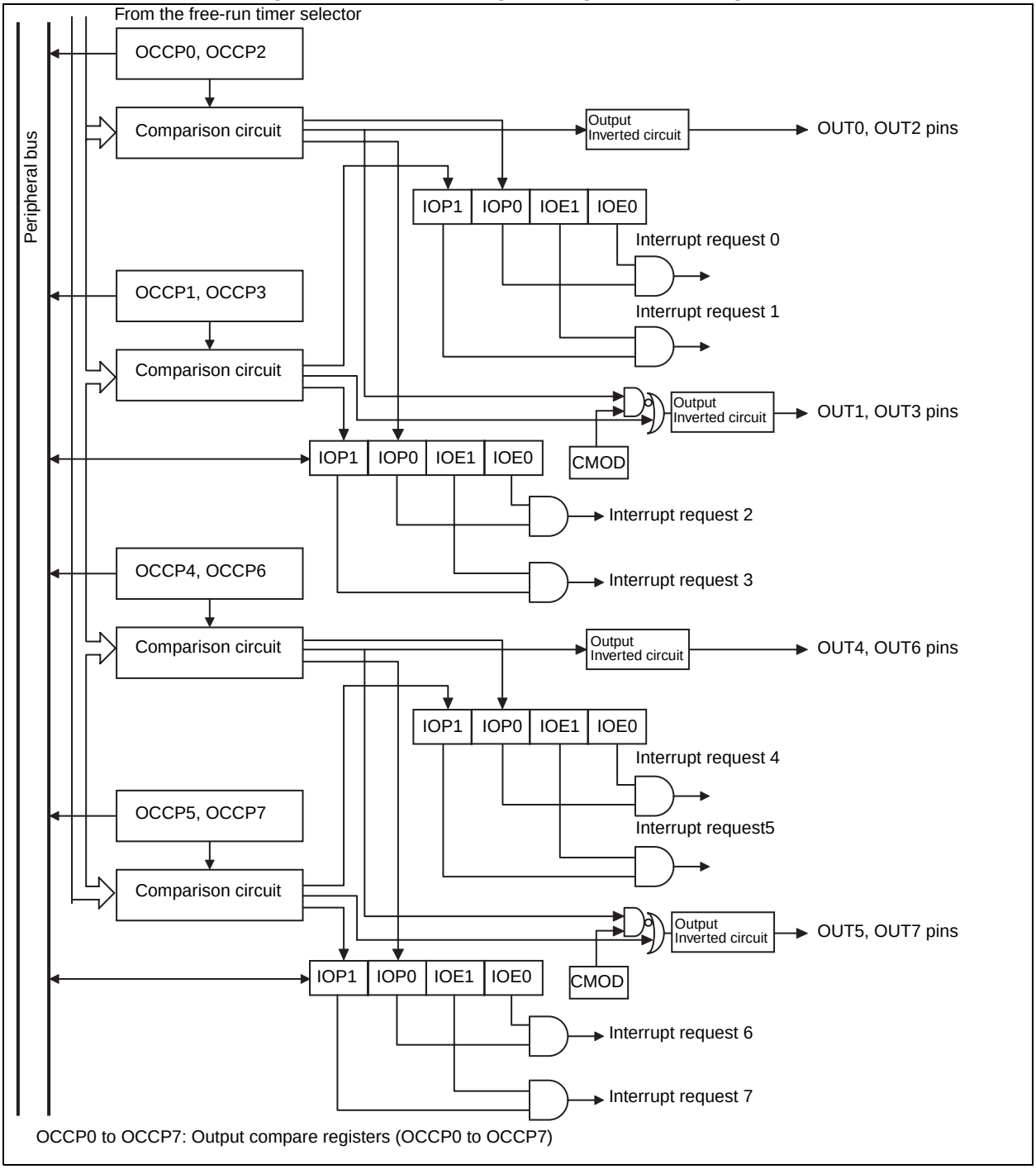
21.2 Configuration

This section explains the configuration of the 32-bit output compare.

■ 32-bit output compare block diagram

Figure 21.2-1 is a block diagram of the 32-bit output compare.

Figure 21.2-1 32-bit output compare block diagram



- Output compare register (OCCP0 to OCCP7)

  This register sets the value (compare value) to be compared with the count value of the 32-bit free-run timer.
- Compare control register

  This register controls the operation of the 32-bit output compare. This register is divided into the following two registers:
  - Compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7)
  - Compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)
- Comparison circuit

  This circuit compares the count value of the 32-bit free-run timer and the compare value that is set in the output compare register (OCCP0 to OCCP7).

<Note>

For details of the compare timer block diagram, see "■ Compare timer block diagram" in "CHAPTER 19 32-bit Free-Run Timer".

■ Clocks

Table 21.2-1 lists the clock used for the 32-bit output compare.

Table 21.2-1 Clock used for 32-bit output compare

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

21.3 Pins

This section explains the pins used by the 32-bit output compare.

■ Overview

- OUT0 to OUT7 pins

  These are the output pins of the 32-bit output compare. These pins are multiplexed pins.

  For details of how to use these pins as the OUT0 to OUT7 pins of the 32-bit output compare, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 21.3-1 lists the relationship between channels and pins.

Table 21.3-1 Relationship between channels and pins

Channel	Output Pin
0	OUT0
1	OUT1
2	OUT2
3	OUT3
4	OUT4
5	OUT5
6	OUT6
7	OUT7

21.4 Registers

This section explains the configuration and functions of the registers used by the 32-bit output compare.

■ 32-bit output compare registers

Table 21.4-1 lists the registers of the 32-bit output compare.

Table 21.4-1 Registers of 32-bit output compare

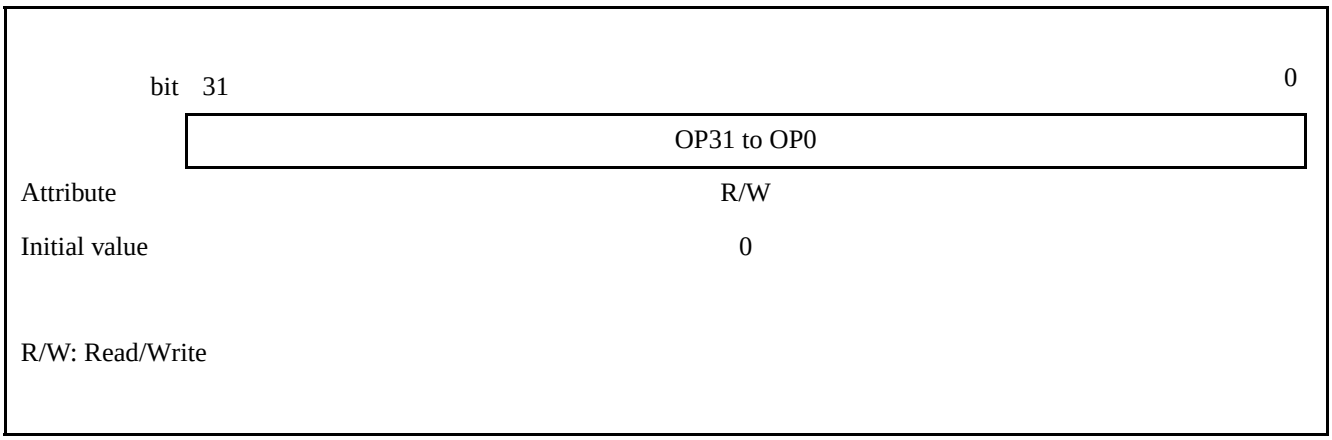
Channel	Abbreviated Register Name	Register Name	Reference
Common	FRTSEL	Free-run timer select register	19.4.1
Common to 0 and 1	OCSH1	Compare control register upper1	21.4.2
	OCSL0	Compare control register lower 0	21.4.3
Common to 2 and 3	OCSH3	Compare control register upper 3	21.4.2
	OCSL2	Compare control register lower 2	21.4.3
Common to 4 and 5	OCSH5	Compare control register upper5	21.4.2
	OCSL4	Compare control register lower 4	21.4.3
Common to 6 and 7	OCSH7	Compare control register upper7	21.4.2
	OCSL6	Compare control register lower 6	21.4.3
0	OCCP0	Output compare register 0	21.4.1
1	OCCP1	Output compare register 1	21.4.1
2	OCCP2	Output compare register 2	21.4.1
3	OCCP3	Output compare register 3	21.4.1
4	OCCP4	Output compare register 4	21.4.1
5	OCCP5	Output compare register 5	21.4.1
6	OCCP6	Output compare register 6	21.4.1
7	OCCP7	Output compare register 7	21.4.1

21.4.1 Output Compare Register (OCCP0 to OCCP7)

This register sets the value (compare value) to be compared with the count value of the 32-bit free-run timer. Set the compare value in this register before activating the 32-bit free-run timer.

Figure 21.4-1 shows the bit configuration of the output compare register (OCCP0 to OCCP7).

Figure 21.4-1 Bit configuration of output compare register (OCCP0 to OCCP7)



<Notes>

- This register can be rewritten even while the 32-bit free-run timer is active.
- The value written to this register is immediately used as a compare value. Therefore, if the compare value is rewritten from a small value to a large value during operation of the 32-bit free-run timer, an interrupt request is generated twice while the 32-bit free-run timer counts once.

  To prevent this problem, rewrite this register by using interrupt processing by the 32-bit free-run timer.
- Be sure to access this register in units of words (32 bits).
- Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is to be compared with the value set in this register varies depending on the free-run timer select register (FRTSEL) setting. For details, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

21.4.2 Compare Control Register Upper (OCSH1, OCSH3, OCSH5, OCSH7)

This register is used to specify whether to use the 2 channels of the 32-bit output compare independently of each other or as a pair. The register is also used to set the level of signals output through the OUT0 to OUT7 pins when the 32-bit output compare function is activated.

Figure 21.4-2 shows the bit configuration of the compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7).

Figure 21.4-2 Bit configuration of compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	CMOD	Undefined	Undefined	OTD1	OTD0
Attribute	-	-	-	R/W	-	-	R/W	R/W
Initial value	X	X	X	0	X	X	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15 to bit13]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit12]: CMOD (output level invert mode bit)

This bit is used to specify whether to use the 2 channels of the 32-bit output compare independently of each other or as a pair. The invert mode of wave forms output from pins changes depending on this setting.

Written Value	Explanation
0	2 channels of the 32-bit output compare are used independently of each other. When the compare value of the output compare register (OCCP0 to OCCP7) matches the count value of the 32-bit free-run timer, the output level from the corresponding pin is inverted.
1	2 channels of the 32-bit output compare are used as a pair. When the compare value of the output compare register (OCCP0 to OCCP7) matches the value of the 32-bit free-run timer, the invert mode is inverted as shown below: When the count value matches the compare value of the even-numbered channel output compare register (OCCP0, OCCP2, OCCP4, OCCP6): the output levels from the following pins are inverted: - Output level from the pin corresponding to the channel - Output level from the pin corresponding to the odd-numbered channel used as a pair. When the count value matches the compare value of the odd-numbered channel output compare register (OCCP1, OCCP3, OCCP5, OCCP7): the output level from the following pin is inverted: - Output level from the pin corresponding to the channel

Table 21.4-2 summarizes the invert timings for output levels from OUT0 to OUT7 pins when "1" is set to this bit.

Table 21.4-2 Output level invert timing

Register Whose Compare Value Matches the Value of the 32-bit Free-run Timer	Pin Whose Output Level Inverts
Output compare register 0 (OCCP0)	OUT0 pin, OUT1 pin
Output compare register 1 (OCCP1)	OUT1 pin
Output compare register 2 (OCCP2)	OUT2 pin, OUT3 pin
Output compare register 3 (OCCP3)	OUT3 pin
Output compare register 4 (OCCP4)	OUT4 pin, OUT5 pin
Output compare register 5 (OCCP5)	OUT5 pin
Output compare register 6 (OCCP6)	OUT6 pin, OUT7 pin
Output compare register 7 (OCCP7)	OUT7 pin

<Notes>

- If the same compare value is set for the even-numbered and odd-numbered channels of the 32-bit output compare, the operation is the same as when the 2 channels of the 32-bit output compare are used independently of each other, even when "1" is set to this bit.
- Be sure to set "1" to this bit when the 2 channels of the 32-bit output compare are used as a pair.

[bit11, bit10]: Reserved bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9]: OTD1 (output level bit)

This bit sets the signal level output from pins (OUT1, OUT3, OUT5, OUT7) when the odd-numbered channel of the 32-bit output compare is activated.

OTD1	In Case of Writing	In Case of Reading
0	The "L" level is output.	The output level is read.
1	The "H" level is output.	

<Note>

Do not rewrite this bit during 32-bit output compare operation.

[bit8]: OTD0 (output level bit)

This bit sets the signal level output from pins (OUT0, OUT2, OUT4, OUT6) when the even-numbered channels of the 32-bit output compare are activated.

OTD0	In Case of Writing	In Case of Reading
0	The "L" level is output.	The output level is read.
1	The "H" level is output.	

<Note>

Do not rewrite this bit during 32-bit output compare operation.

21.4.3 Compare Control Register Lower (OCSL0, OCSL2, OCSL4, OCSL6)

This register enables or disables 32-bit output compare operation or controls interrupt requests.

Figure 21.4-3 shows the bit configuration of the compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6).

Figure 21.4-3 Bit configuration of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

bit	7	6	5	4	3	2	1	0
	IOP1	IOP0	IOE1	IOE0	Undefined	Undefined	CST1	CST0
Attribute	R/W	R/W	R/W	R/W	–	–	R/W	R/W
Initial value	0	0	0	0	X	X	0	0

R/W: Read/Write

–: Undefined

X: Undefined

[bit7]: IOP1 (odd-numbered channel compare match interrupt request flag bit)

This bit indicates that the compare value of the odd-numbered channel output compare register (OCCP1, OCCP3, OCCP5, OCCP7) matches the count value of the 32-bit free-run timer.

If "1" is set to the IOE1 bit when this bit is "1", a compare match interrupt request is generated.

IOP1	In Case of Reading	In Case of Writing
0	A comparison result indicates no match.	This bit is cleared to "0".
1	A comparison result indicates a match.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit6]: IOP0 (even-numbered channel compare match interrupt request flag bit)

This bit indicates that the compare value of the even-numbered channel output compare register (OCCP0, OCCP2, OCCP4, OCCP6) matches the count value of the 32-bit free-run timer.

If "1" is set to the IOE0 bit when this bit is "1", a compare match interrupt request is generated.

IOP0	In Case of Reading	In Case of Writing
0	A comparison result indicates no match.	This bit is cleared to "0".
1	A comparison result indicates a match.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit5]: IOE1 (odd-numbered channel compare match interrupt enable bit)

This bit specifies whether to generate a compare match interrupt request when the value of the odd-numbered channel output compare register (OCCP1, OCCP3, OCCP5, OCCP7) matches the count value of the 32-bit free-run timer (IOP1=1).

Written Value	Explanation
0	Disables generation of compare match interrupt requests.
1	Enables generation of compare match interrupt requests.

[bit4]: IOE0 (even-numbered channel compare match interrupt enable bit)

This bit specifies whether to generate a compare match interrupt request when the value of the even-numbered channel output compare register (OCCP0, OCCP2, OCCP4, OCCP6) matches the count value of the 32-bit free-run timer (IOP0=1).

Written Value	Explanation
0	Disables generation of compare match interrupt requests.
1	Enables generation of compare match interrupt requests.

[bit3, bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1]: CST1 (odd-numbered channel compare enable bit)

This bit enables or disables the comparison between odd-numbered channel 32-bit output compare and the count value of the 32-bit free-run timer.

Written Value	Explanation
0	Disables comparison.
1	Enables comparison.

<Note>

When the 32-bit free-run timer is stopped, the comparison of 32-bit output compare is also stopped.

[bit0]: CST0 (even-numbered channel compare enable bit)

This bit enables or disables the comparison between even-numbered channel 32-bit output compare and the count value of the 32-bit free-run timer.

Written Value	Explanation
0	Disables comparison.
1	Enables comparison.

<Note>

When the 32-bit free-run timer is stopped, the comparison of 32-bit output compare is also stopped.

21.5 Interrupts

An interrupt request (compare match interrupt request) is generated when the count value of the 32-bit free-run timer matches the value set in the output compare register (OCCP0 to OCCP7).

Table 21.5-1 outlines the interrupts that can be used with the 32bit output compare.

Table 21.5-1 Interrupts of the 32-bit output compare

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Compare result match interrupt request	Even-numbered channel: IOP0=1 for OCSL Odd-numbered channel: IOP1=1 for OCSL	Even-numbered channel: IOE0=1 for OCSL Odd-numbered channel: IOE1=1 for OCSL	Write "0" to the next bit Even-numbered channel: IOP0 bit for OCSL Odd-numbered channel: IOP1 bit for OCSL

OCSL: compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use an interrupt control register (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

21.6 An Explanation of Operations and Setting Procedure Examples

This section explains the 32-bit output compare operation. Also, examples of procedures for setting the operating state are shown.

■ Overview

2 channels of the 32-bit output compare can be used either independently of each other or as a pair.

21.6.1 When the 2 Channels Are Used Independently of Each Other

This section explains the 32-bit output compare operation when the 2 channels are used independently of each other.

■ Overview

When the CMOD bit of the compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7) is set to "0", the 2 channels of the 32-bit output compare operate independently of each other.

The output level of the pin corresponding to the channel is inverted when the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP7).

<Note>

Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is to be compared with the value set in the output compare register (OCCP0 to OCCP7) varies depending on the free-run timer select register (FRTSEL) setting. For details, see " 19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

■ Operation

Writing "1" to the following bit enables the 32-bit output compare operation.

- Enabling even-numbered channel operation: CST0 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)
- Enabling odd-numbered channel operation: CST1 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

When the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP7) while the 32-bit output compare is enabled, the following bits are set to "1":

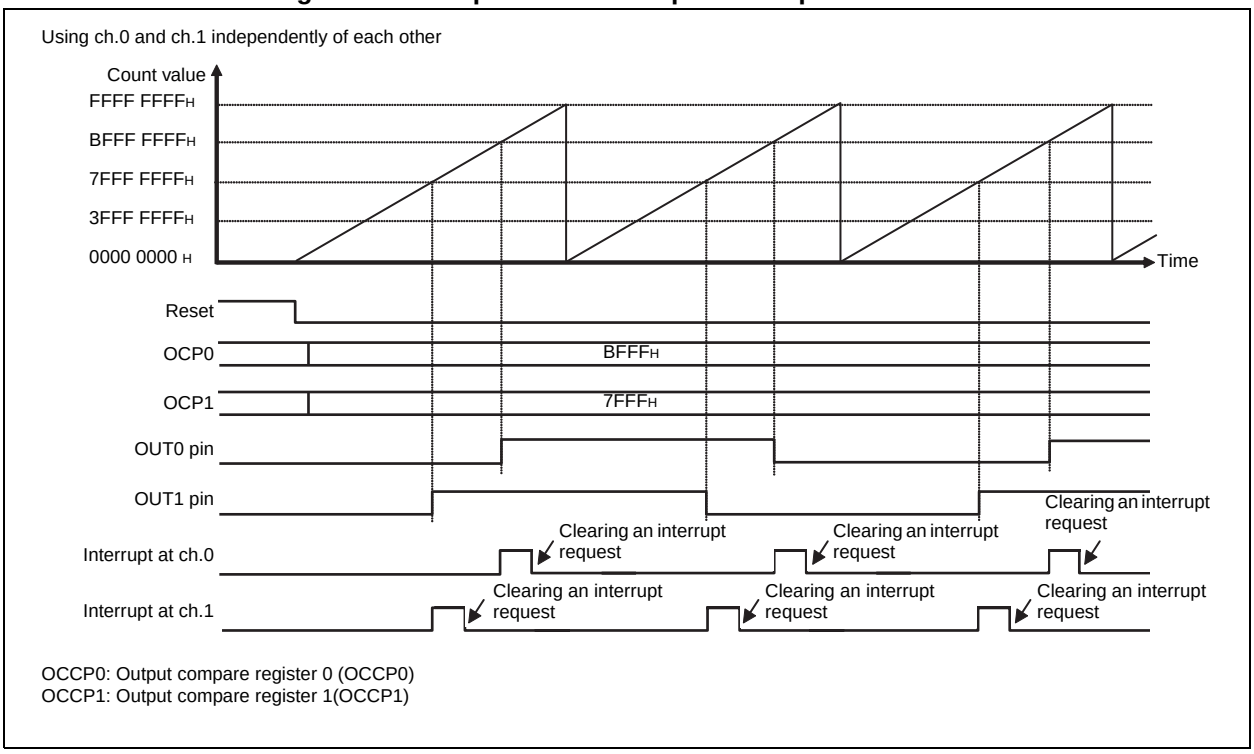
- Even-numbered channel: IOP0 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)
- Odd-numbered channel: IOP1 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

If interrupt request generation has been enabled, a compare match interrupt request is generated.

Also, the output levels from the OUT0 to OUT7 pins are inverted.

Figure 21.6-1 shows the operation in independent operation mode.

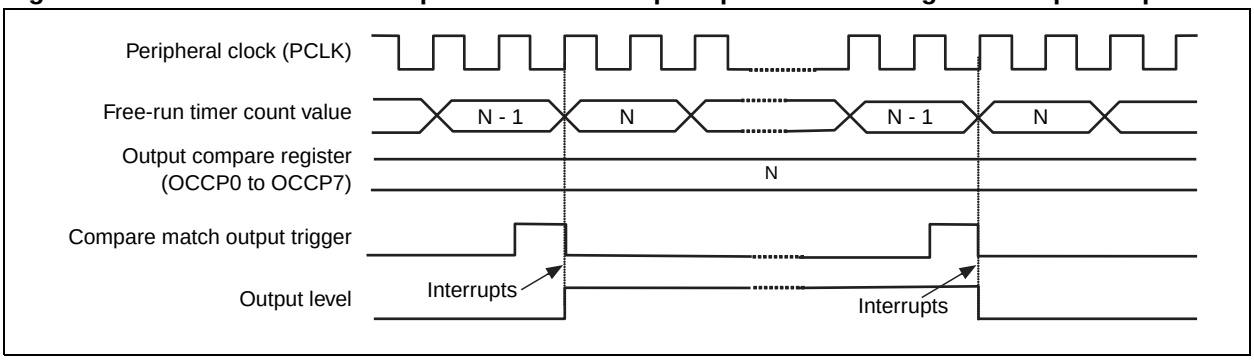
Figure 21.6-1 Operation in independent operation mode



A compare match interrupt request or a change in the pin output level occurs upon detection of a compare match.

Figure 21.6-2 shows the generation of compare match interrupt requests and changes in the pin output level.

Figure 21.6-2 Generation of compare match interrupt requests and changes in the pin output level



<Note>

When using 2 channels of the 32-bit output compare independently of each other, be sure to write "0" to the CMOD bit of the compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7).

21.6.2 When the 2 Channels Are Used as a Pair

This section explains the 32-bit output compare operation using the even-numbered and odd-numbered channels in pairs.

■ Overview

When the CMOD bit of the compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7) is set to "1", the 2 channels of the 32-bit output compare operate in pairs.

By using the even-numbered and odd-numbered channels of the 32-bit output compare in pairs, compare values for 2 channels can be updated by 1 interrupt.

The combinations of even-numbered and odd-numbered channels that can be used in pairs are as follows:

- ch.0 and ch.1
- ch.2 and ch.3
- ch.4 and ch.5
- ch.6 and ch.7

■ Operation

Writing "1" to the following bit enables the 32-bit output compare operation.

- Enabling even-numbered channel operation: CST0 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)
- Enabling odd-numbered channel operation: CST1 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

When the count value of the 32-bit free-run timer matches the compare value of the output compare register (OCCP0 to OCCP7) while the 32-bit output compare is enabled, the following bits are set to "1":

- Even-numbered channel: IOP0 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)
- Odd-numbered channel: IOP1 bit of compare control register lower (OCSL0, OCSL2, OCSL4, OCSL6)

If interrupt request generation has been enabled, a compare match interrupt request is generated.

Also, the output levels from the OUT0 to OUT7 pins are inverted. The pin whose output level is inverted varies depending on the channel of the output compare register (OCCP0 to OCCP7) whose compare value matches the count value of the 32-bit free-run timer.

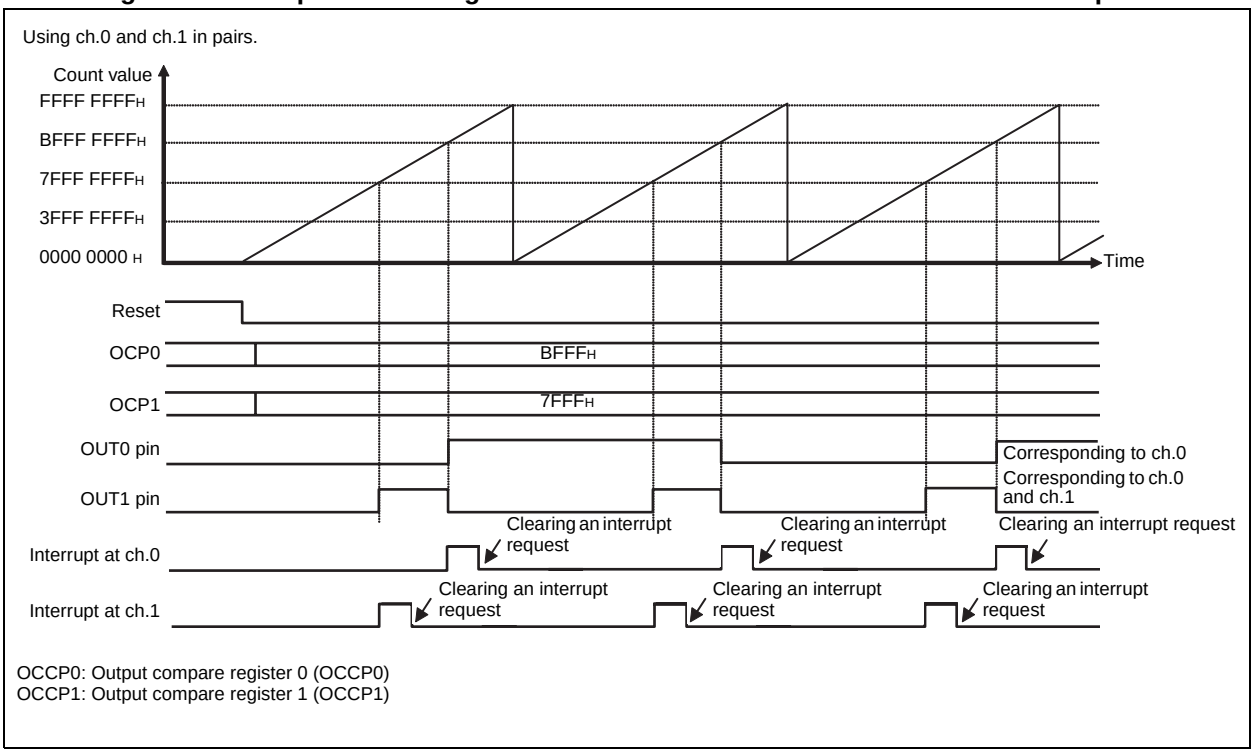
Table 21.6-1 shows the relationship between the channels for which compare values are set and the pins whose output levels are inverted.

Table 21.6-1 Relationship between the channels for which compare values are set and the pins whose output levels are inverted

Register Whose Compare Value Matches the Value of the 32-bit Free-run Timer	Pin Whose Output Level Inverts
Output compare register 0 (OCCP0)	OUT0 pin, OUT1 pin
Output compare register 1 (OCCP1)	OUT1 pin
Output compare register 2 (OCCP2)	OUT2 pin, OUT3 pin
Output compare register 3 (OCCP3)	OUT3 pin
Output compare register 4 (OCCP4)	OUT4 pin, OUT5 pin
Output compare register 5 (OCCP5)	OUT5 pin
Output compare register 6 (OCCP6)	OUT6 pin, OUT7 pin
Output compare register 7 (OCCP7)	OUT7 pin

Figure 21.6-3 shows the operation using even-numbered and odd-numbered channels in pairs.

Figure 21.6-3 Operation using even-numbered and odd-numbered channels in pairs



A compare match interrupt request or a change in the pin output level occurs upon detection of a compare match.

See "21.6.1 When the 2 Channels Are Used Independently of Each Other" for details of the generation of compare match interrupt requests and changes in the pin output level.

<Notes>

- When using even-numbered and odd-numbered channels of the 32-bit output compare in pairs, be sure to write "1" to the CMOD bit of the compare control register upper (OCSH1, OCSH3, OCSH5, OCSH7).
- Of the 2 channels of the 32-bit free-run timer, the channel of the 32-bit free-run timer whose value is to be compared with the value set in the output compare register (OCCP0 to OCCP7) varies depending on the free-run timer select register (FRTSEL) setting. For details, see "19.4.1 Free-Run Timer Select Register (FRTSEL)" in "CHAPTER 19 32-bit Free-Run Timer".

CHAPTER 22 16-bit Reload Timer

This chapter explains the functions and operations of the 16-bit reload timer.

22.1 Overview
22.2 Configuration
22.3 Pins
22.4 Registers
22.5 Interrupts
22.6 An Explanation of Operations and Setting Procedure Examples
22.7 Notes on Use

22.1 Overview

The 16-bit reload timer is a down counter that performs a countdown from a preset value. This timer can be used as an interval timer that counts down synchronously with an internal clock (peripheral clock), and it can also be used as an event counter that counts external events.
This series has 3 built-in channels of the 16-bit reload timer.

■ Overview

- Timer mode: Internal timer mode and event counter mode are available.
  - Interval timer mode

    It counts down synchronously with an internal clock (peripheral clock). The internal clock (peripheral clock) is selected from 6 clock types, which are peripheral clocks (PCLK) divided by 2, 4, 8, 16, 32, and 64.
  - Event counter mode

    It detects and counts the edges (rising edge/falling edge/both edges) of the external clock.

    Cascade mode that counts ch.0 outputs with ch.1 and ch.1 outputs with ch.2 is also available.
- Operation mode: One of the following two modes can be selected.
  - Reload mode

    In this mode, the reload value is reloaded, and counting is repeated when the down counter enters an underflow condition.
  - One shot mode

    In this mode, counting stops when the down counter enters an underflow condition.
- Input pin function: In interval timer mode, the trigger input function or gate input function can be selected for the input pin function.
  - Trigger input function

    When it detects a valid edge (rising edge/falling edge/both edges) from the input pin, it starts counting.
  - Gate input function

    It continues counting as long as the input pin maintains its effective level of input.
- Interrupt request: It can generate an interrupt request when the down counter enters an underflow condition.

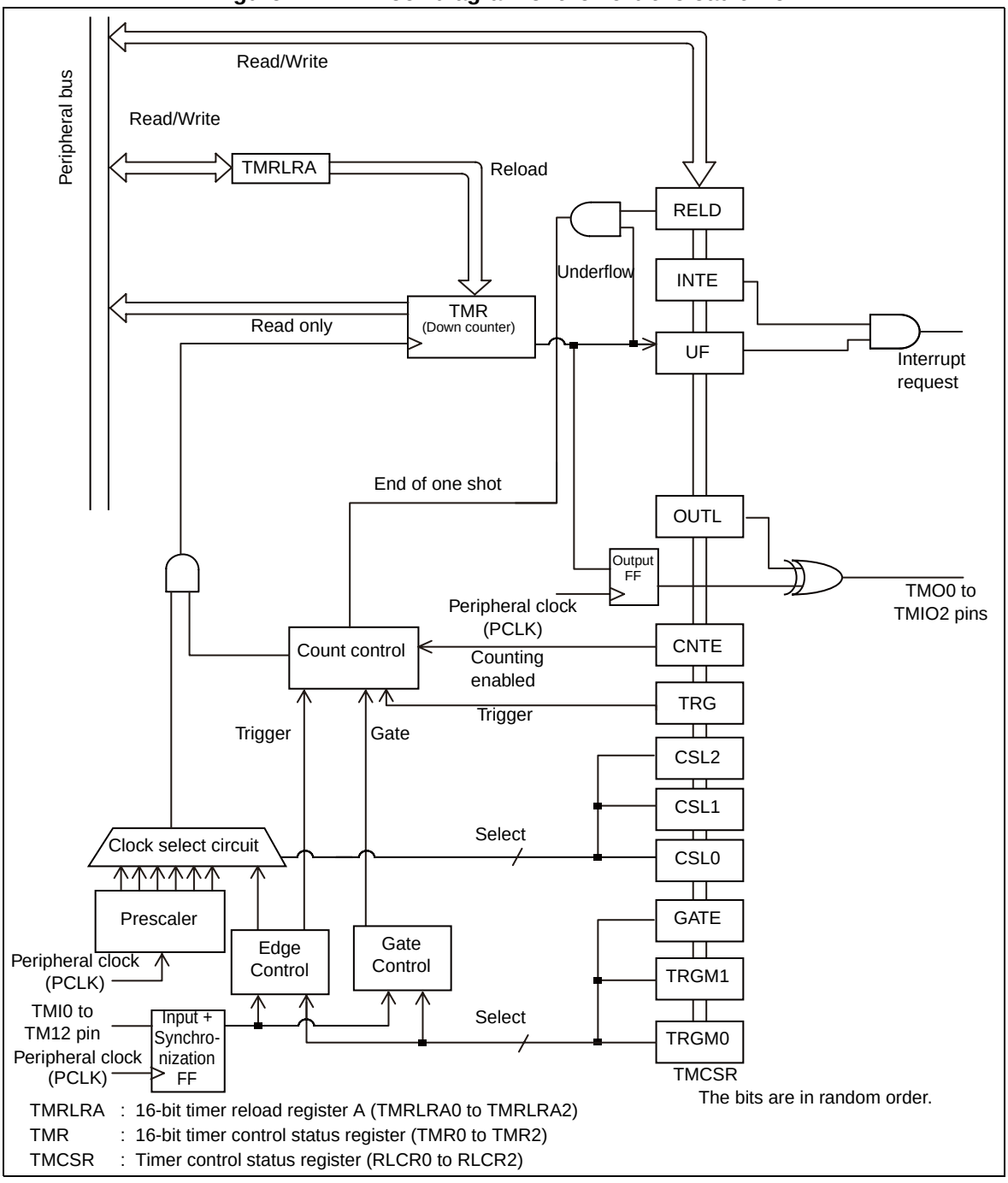
22.2 Configuration

This section explains the 16-bit reload timer configuration.

■ Block diagram of the 16-bit reload timer

Figure 22.2-1 is a block diagram of the 16-bit reload timer.

Figure 22.2-1 Block diagram of the 16-bit reload timer



- Timer control status register (TMCSR0 to TMCSR2)

  This register controls the operations of the 16-bit reload timer.
- 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

  This register sets the reload values.
- 16-bit timer register (TMR0 to TMR2)

  This register operates as a down counter. When this register is read, the down counter value can be read.
- Prescaler

  When the interval timer mode is selected, the prescaler divides the peripheral clock (PCLK).
- Clock select circuit

  The clock select circuit selects a count clock.
- Edge controller

  The edge controller controls the detection edges of signals when the TMI0 to TMI2 pins are used as trigger input pins.
- Gate controller

  The gate controller controls the signal levels of the signals input from the pins when the TMI0 to TMI2 pins are used as gate input pins.
- Count controller

  The count controller controls the counts of the 16-bit reload timer.

■ Clocks

Table 22.2-1 shows the clock used for the 16-bit reload timer.

Table 22.2-1 Clock used for the 16-bit reload timer

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Created through division of the peripheral clock (PCLK).
	External clock	Input from TMI0 to TMI2 pins

22.3 Pins

This section explains the pins of the 16-bit reload timer.

■ Overview

There are two types of 16-bit reload timer as follows.

- TMO0 to TMO2 pins

  16-bit reload timer wave form output pin

  These pins are multiplexed pins. For information on using as the wave form output pin of the 16-bit reload timer, see "2.4 Setting Method for Pins".

- TMI0 to TMI2 pins

  16-bit reload timer input pin This inputs count clock, clock, trigger, or gate depending on its setting.

  These pins are multiplexed pins. For information on using as the input pin of the 16-bit reload timer, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 22.3-1 outlines the relationship between channels and pins.

Table 22.3-1 Relationship between channels and pins

Channel	Wave Form Output Pin	Input Pin
0	TMO0	TMI0
1	TMO1	TMI1
2	TMO2	TMI2

22.4 Registers

This section explains the configuration and functions of registers used by the 16-bit reload timer.

■ Registers of 16-bit reload timer

Table 22.4-1 lists the registers of the 16-bit reload timer.

Table 22.4-1 Registers of 16-bit reload timer

Channel	Abbreviated Register Name	Register Name	Reference
0	TMCSR0	Timer control status register 0	22.4.1
	TMRLRA0	16-bit timer reload register A0	22.4.2
	TMR0	16-bit timer register 0	22.4.3
1	TMCSR1	Timer control status register 1	22.4.1
	TMRLRA1	16-bit timer reload register A1	22.4.2
	TMR1	16-bit timer register 1	22.4.3
2	TMCSR2	Timer control status register 2	22.4.1
	TMRLRA2	16-bit timer reload register A2	22.4.2
	TMR2	16-bit timer register 2	22.4.3

22.4.1 Timer Control Status Register (TMCSR0 to TMCSR2)

This register controls the operations of the 16-bit reload timer.

Figure 22.4-1 shows the bit configuration of the timer control status registers (TMCSR0 to TMCSR2).

Figure 22.4-1 Bit configuration of the timer control status registers (TMCSR0 to TMCSR2)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	TRGM1	TRGM0	CSL2	CSL1	CSL0	GATE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	OUTL	RELD	INTE	UF	CNTE	TRG
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit15, bit14]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: TRGM1, TRGM0 (Input pin operation selection bit)

This bit selects the operation of TMI0 to TMI2 pins of the 16-bit reload timer.The meaning of this bit varies depending whether the 16-bit reload timer is used in interval timer mode, or in event counter mode.

- Interval timer mode (CSL2 to CSL0 = 000 to 101)
  - Select the trigger input function with TMI0 to TMI2 pins (GATE = 0).

    Select an effective edge.

    When the edge set with this bit is detected in the signal input from the TMI0 to TMI2 pins, the down counter starts counting down.
  - Select the gate function with TMI0 to TMI2 pins (GATE = 1).

    Select an effective level.

    The down counter counts down only while the signal of the level that is set with this bit is input from the TMI0 to TMI2 pins.

TRGM1	TRGM0	When the Trigger Input Is Selected * (GATE =0)	When the Gate Function Is Selected (GATE =1)
0	0	Edge detection disabled	"L" level
0	1	Rising edge	"H" level
1	0	Falling edge	"L" level
1	1	Both edges	"H" level

* When "1" is written in the TRG bit, the down counter starts counting down regardless of the setting of this bit.

- In event counter mode (CSL2 to CSL0 = 110, 111)

  Select an effective edge.

  When the edge set with this bit is detected in the signal input from the TMI0 to TMI2 pins, the down counter starts counting down.

TRGM1	TRGM0	Explanation
0	0	Setting prohibited
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

<Note>

Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit11 to bit9]: CSL2 to CSL0 (Count source selection bits)

This bit selects the timer mode of the 16-bit reload timer. In interval timer mode, it also selects the division rate of the peripheral clock (PCLK), and in event counter mode, it also selects whether to use cascade mode and whether to use the external clock.

CSL2	CSL1	CSL0	Explanation	
0	0	0	Interval timer mode	Peripheral clock (PCLK) divided by 2 (= 2^1)
0	0	1		Peripheral clock (PCLK) divided by 4 (= 2^2)
0	1	0		Peripheral clock (PCLK) divided by 8 (= 2^3)
0	1	1		Peripheral clock (PCLK) divided by 16 (= 2^4)
1	0	0		Peripheral clock (PCLK) divided by 32 (= 2^5)
1	0	1		Peripheral clock (PCLK) divided by 64 (= 2^6)
1	1	0	Event counter mode	Cascade mode*
1	1	1		External clock

* For information on the operation when cascade mode is selected, see "22.6.3 Operation in Cascade Mode".

<Notes>

- Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

  If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.
- To use the 2-channel 16-bit reload timer connected in cascade, set this bit as shown below.
  - Channel with smaller number: Select interval timer mode or an external clock.
  - Channel with larger number: Specify cascade mode.
- When event counter mode is selected for this bit, the setting of the GATE bit is ignored.

[bit8]: GATE (Gate input enable bit)

When the timer mode is set to interval timer mode, this bit selects the functions to be assigned to the TMI0 to TMI2 pins.

- Trigger input function: When an effective edge is input from TMI0 to TMI2 pins, a countdown starts.
- Gate function: A countdown is performed only while the effective level signal is input from TMI0 to TMI2 pins.

Written Value	Explanation
0	Trigger input function
1	Gate function

<Notes>

- Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

  If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.
- If event counter mode is selected with CSL2 to CSL0 bits (CSL2 to CSL0 = 110/111), this bit setting is ignored.

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit5]: OUTL (Output polarity setting bit)

When the 16-bit reload timer is activated, this bit sets the signal level of the signals to be output from TMO0 to TMO2 pins.

Written Value	Explanation
0	Normal polarity ("L" level)
1	Inverted polarity ("H" level)

<Note>

Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit4]: RELD (Reload operation enable bit)

This bit selects any of the following operation modes for the 16-bit reload timer.

- One shot mode

  When the down counter enters an underflow condition, counting stops in this mode until the next activation trigger is input.

- Reload mode

  When the down counter enters an underflow condition in this mode, the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter so that it continues counting.

Written Value	Explanation
0	One shot mode
1	Reload mode

<Note>

Be sure to change this bit after operation of the down counter is stopped by the CNTE bit (CNTE = 0).

If this bit is changed at the same time that the CNTE bit is changed, this bit is changed regardless of the value of the CNTE bit.

[bit3]: INTE (Interrupt request enable bit)

This bit sets whether to generate the underflow interrupt request when the down counter underflows (UF bit = 1).

Written Value	Explanation
0	Disables generation of underflow interrupt requests.
1	Enables generation of underflow interrupt requests.

[bit2]: UF (Underflow interrupt request flag bit)

This bit indicates that the down counter enters an underflow condition.

If the INTE is set to "1" when this bit is "1", an underflow interrupt request is generated.

UF	In Case of Reading	In Case of Writing
0	The down counter has not entered an underflow condition.	This bit is cleared to "0".
1	The down counter has entered an underflow condition.	Ignored

[bit1]: CNTE (Count operation enable bit)

This bit enables/disables the operation of the down counter.

Written Value	Explanation
0	Stops the count operation.
1	Enables the count operation (activation trigger wait).

<Note>

If "0" is written to this bit during a down counter operation, the down counter stops.

[bit0]: TRG (Software trigger bit)

This bit activates the 16-bit reload timer through software. When "1" is written to this bit, the down counter loads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and starts counting.

TRG	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates the 16-bit reload timer.	

<Notes>

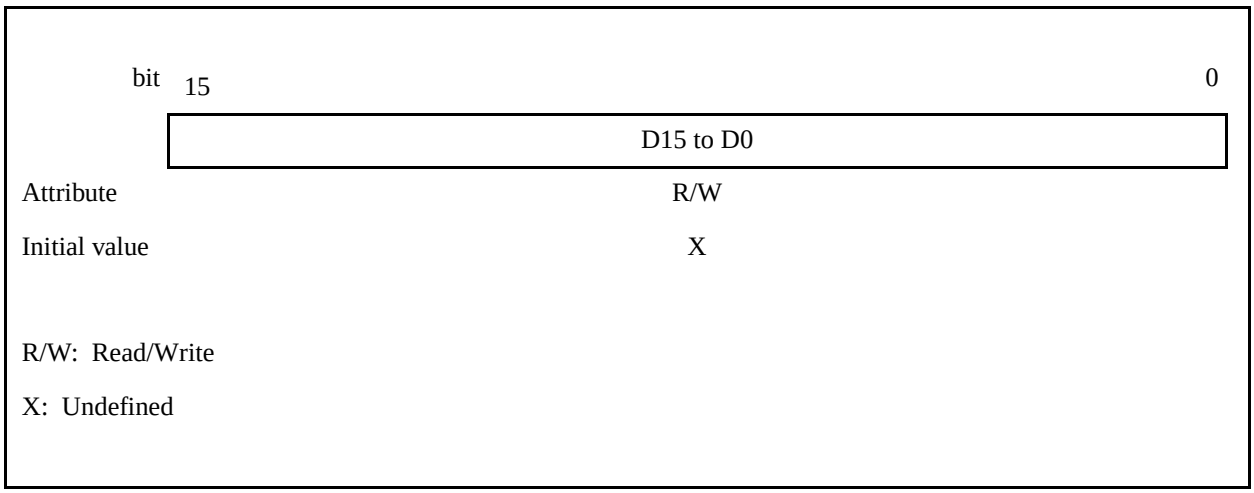
- The down counter does not operate while the CNTE bit is "0" even if "1" is written to this bit.
- When the 16-bit reload timer operation is enabled (CNTE=1), if "1" is written to this bit, the down counter starts regardless of the setting of TRGM1 or TRGM0 bit.

22.4.2 16-bit Timer Reload Register A (TMRLRA0 to TMRLRA2)

This register sets the initial value of the down counter.
In reload mode, if an underflow occurs, the value of this register is reloaded to the down counter.

Figure 22.4-2 shows the bit configuration of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

Figure 22.4-2 Bit configuration of 16-bit timer reload register A (TMRLRA0 to TMRLRA2)



When the counter completes counting the value set to this register + 1, an underflow occurs.The signal level of the signals output from TMO0 to TMO2 pins is inverted.

<Note>

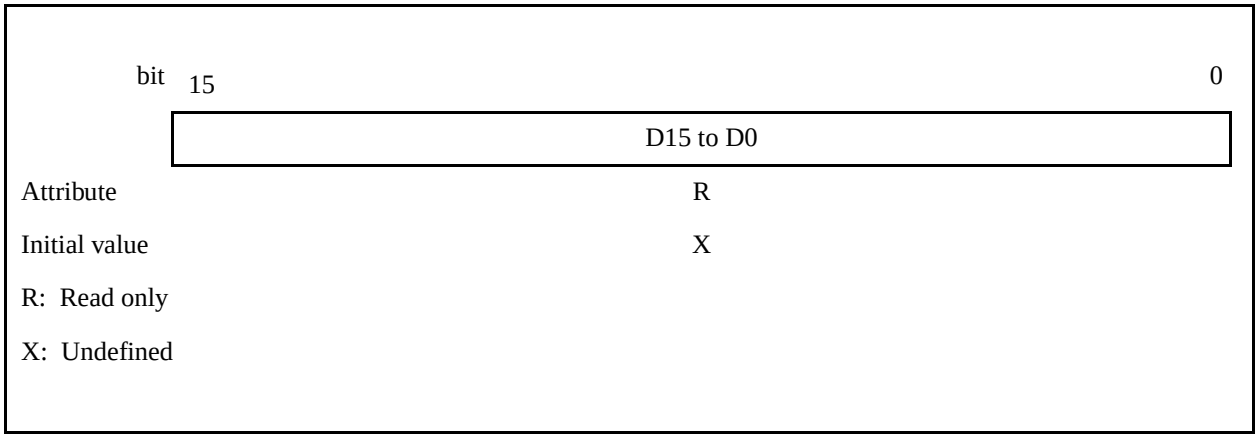
Be sure to access this register in units of half words.

22.4.3 16-bit Timer Register (TMR0 to TMR2)

When this register is read, the down counter value can be read.

Figure 22.4-3 shows the bit configuration of the 16-bit timer registers (TMR0 to TMR2).

Figure 22.4-3 Bit configuration of 16-bit timer register (TMR0 to TMR2)



<Note>

Be sure to read this register in units of half words.

22.5 Interrupts

An underflow interrupt request is generated when the down counter enters an underflow condition.

■ Overview

Table 22.5-1 outlines the interrupts that can be used with the 16-bit reload timer

Table 22.5-1 Interrupts of the 16-bit reload timer

Interrupt request	Interrupt request flag	Interrupt request enabled	Clearing an interrupt request
Underflow interrupt request	UF=1 for TMCSR	INTE=1 for TMCSR	Write "0" to the UF bit for TMCSR

TMCSR: timer control status register (TMCSR0 to TMCSR2)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For information on the settings of the interrupt levels, see "CHAPTER 10 Interrupt Controller".

22.6 An Explanation of Operations and Setting Procedure Examples

This chapter explains the operations of the 16-bit reload timer. Also, examples of procedures for setting the operating state are shown.

■ Overview

The 16-bit reload timer is a down counter that counts down from a preset value.One of the following timer modes can be selected using the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2).

- Interval timer mode (CSL2 to CSL0 = 000 to 101)
  It operates with the count clock, which is the divided peripheral clock (PCLK).
- Event counter mode (CSL2 to CSL0 = 110, 111)
  In this mode, the counter counts every time an effective edge is input from TMI0 to TMI2 pins.
  Cascade mode that counts ch.0 outputs with ch.1 and ch.1 outputs with ch.2 is also available.

■ How to set the signal level of the signals output from TMO0 to TMO2 pins.

The signal level of the signals output from TMO0 to TMO2 pins varies with the settings of OUTL bit of the timer control status register (TMCSR0 to TMCSR2).

● In reload mode

Table 22.6-1 shows the signal level of the signals output from TMO0 to TMO2 pins in reload mode.

Table 22.6-1 Signal level in reload mode

	Normal polarity (OUTL = 0)	Inverted polarity (OUTL = 1)
When the 16-bit reload timer is activated	"L" level	"H" level
Subsequent	The output level is inverted every time an underflow occurs.	

● In one shot mode

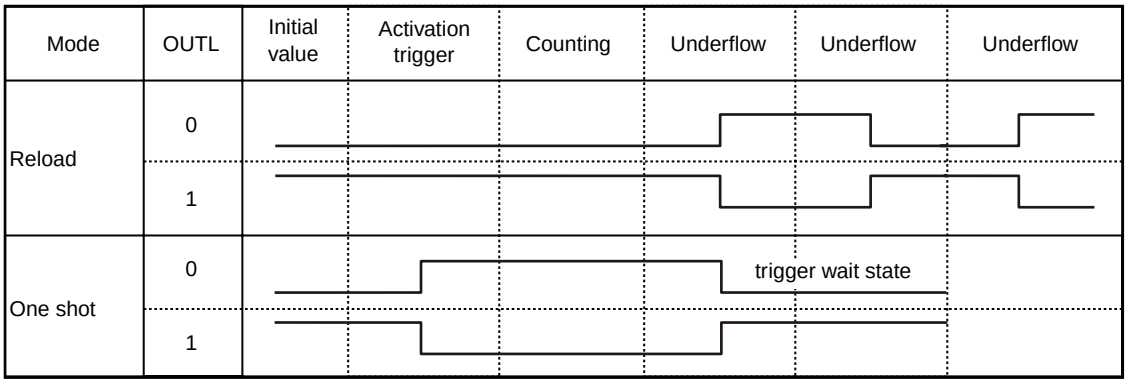
Table 22.6-2 shows the signal level of the signals output from TMO0 to TMO2 pins in one shot mode.

Table 22.6-2 Signal level in one shot mode

	Normal polarity (OUTL = 0)	Inverted polarity (OUTL = 1)
When the 16-bit reload timer is activated	"L" level	"H" level
When an activation trigger is input	"H" level	"L" level
When an underflow occurs	"L" level	"H" level

Figure 22.6-1 shows the OUTL bits of the timer control status register (TMCSR0 to TMCSR2) and their output wave forms.

Figure 22.6-1 OUTL bits of the timer control status registers (TMCSR0 to TMCSR2) and their output wave forms



22.6.1 Operation in Interval Timer Mode

This section explains the operation for using the 16-bit reload timer that counts synchronously with the internal clock (peripheral clock) in interval timer mode.
The count clock is generated by dividing the peripheral clock (PCLK).

■ Setting

This section also explains the settings required for using the 16-bit reload timer in interval timer mode.

● Interval timer mode settings

To use the 16-bit reload timer in interval timer mode, make any of the following settings for the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2), and select the division rate of the peripheral clock (PCLK).

CSL2	CSL1	CSL0	Timer Mode	Division Rate of Peripheral Clock
0	0	0	Interval timer mode	Divided by 2 (= 2^1)
0	0	1		Divided by 4 (= 2^2)
0	1	0		Divided by 8 (= 2^3)
0	1	1		Divided by 16 (= 2^4)
1	0	0		Divided by 32 (= 2^5)
1	0	1		Divided by 64 (= 2^6)

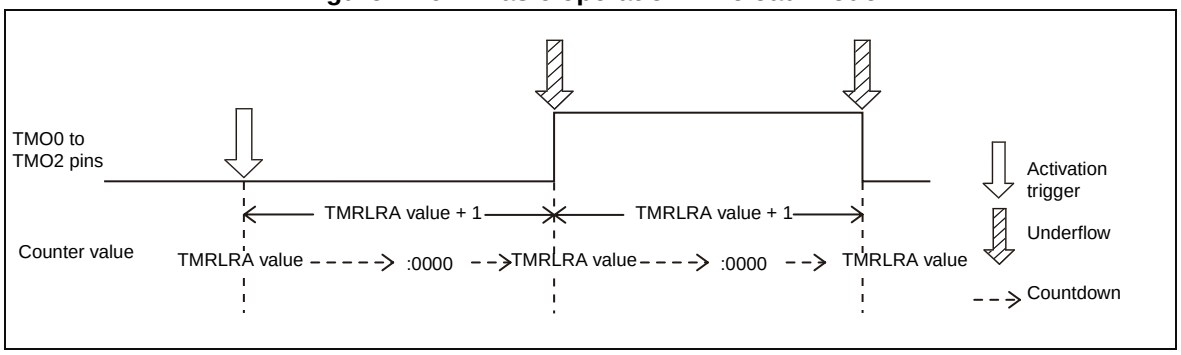
● Operation mode settings

In interval timer mode, one of the following operation modes can be selected using the RELD bits of the timer control status register (TMCSR0 to TMCSR2).

- Reload mode (RELD = 1)

  When the down counter enters an underflow condition, it reloads the value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and repeats counting in this mode. Figure 22.6-2 shows the basic operation in reload mode.

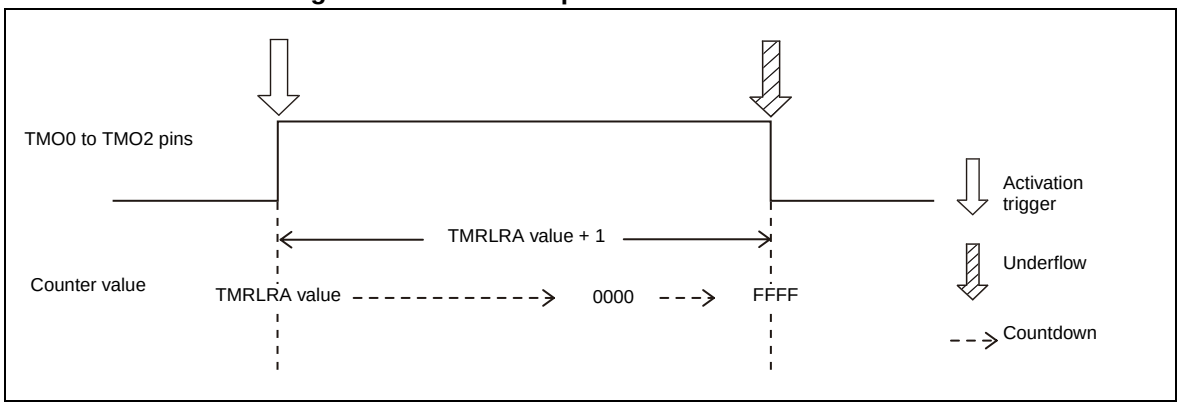
Figure 22.6-2 Basic operation in reload mode



- One shot mode (RELD = 0)

  In this mode, counting stops when the down counter enters an underflow condition. Figure 22.6-3 shows the basic operation in one shot mode.

Figure 22.6-3 Basic operation in one shot mode



● TMI0 to TMI2 pin function settings

Using TRGM1 and TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) and the GATE bit, the function of TMI0 to TMI2 pins can be selected from the following list.

Table 22.6-3 shows the combination of bits.

Table 22.6-3 Combination of bits

TRGM1, TRGM0	GATE	Pin Function
00	0	TMI0 to TMI2 pins do not work.
01	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is a rising edge.
10	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is a falling edge.
11	0	TMI0 to TMI2 pins operate as the trigger input function. The effective edge is both edges.
00/10	1	TMI0 to TMI2 pins operate as the gate input function. The effective level is "L".
01/11	1	TMI0 to TMI2 pins operate as the gate input function. The effective level is "H".

■ Pulse width calculation

How to calculate the pulse width of the signals output from TMO0 to TMO2 pins in interval timer mode is explained below.

Pulse width = T × (L + 1)

L Value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

T Cycles of the count clock

■ How to calculate underflow cycles

If the down counter attempts to count further from the value of "0000_H", an underflow occurs. A cycle from when the down counter starts counting to when an underflow occurs is set in the 16-bit timer reload register (TMRLRA0 to TMRLRA2).

The following shows how to calculate the underflow cycles.

T × (L + 1)

T Cycles of the count clock

L Value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)

In this mode, TMI0 to TMI2 pins are used for trigger input, and the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded each time underflow occurs to continue a countdown.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- One of the TRGM1, TRGM0 bits = 01 to 11
- GATE bit = 0
- RELD bit = 1

● Activate

Use the following procedure for activating.

1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).

   The 16-bit reload timer enters the activation trigger wait state.

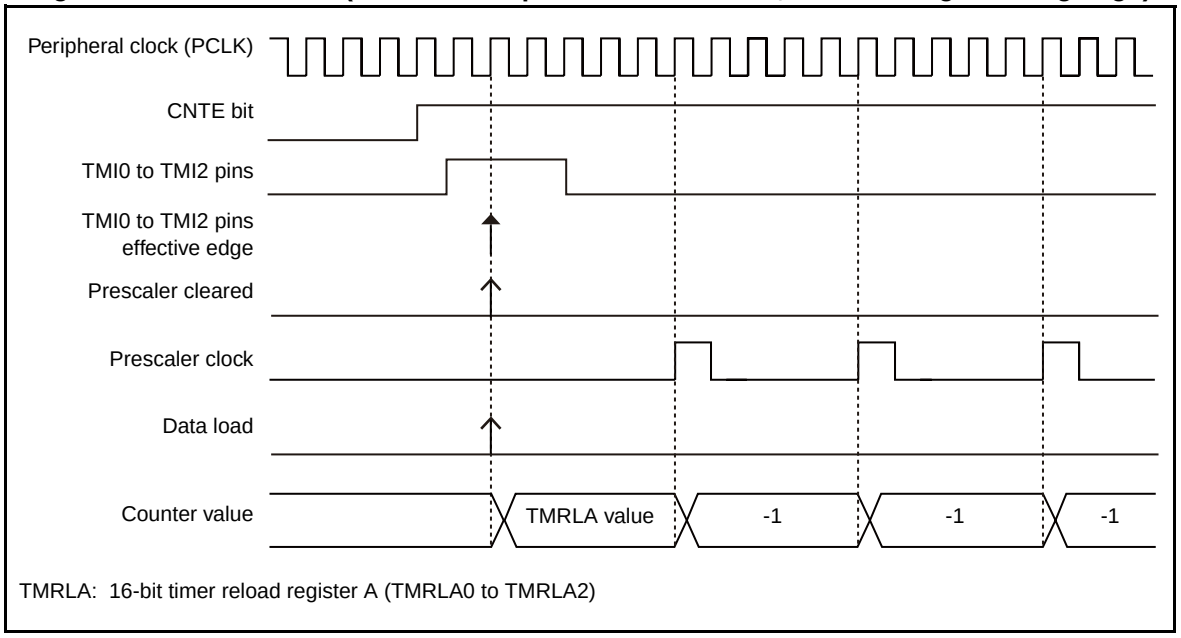
2. The activation trigger is input in either of the following ways:

   - Input the edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.
   - Write "1" to the TRG bit of the timer control status register (TMCSR0 to TMCSR2).

   The prescaler is cleared. The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and counting starts.

Figure 22.6-4 shows an activation.

Figure 22.6-4 Activation (TMI0 to TMI2 pins at an activation, effective edge = rising edge)



<Note>

Be sure that the pulse width of the activation trigger input from TMI0 to TMI2 pins never falls below 2T (T: cycle of the peripheral clock (PCLK)).

● Count operation

The down counter starts a countdown synchronously with the count clock from the value of 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

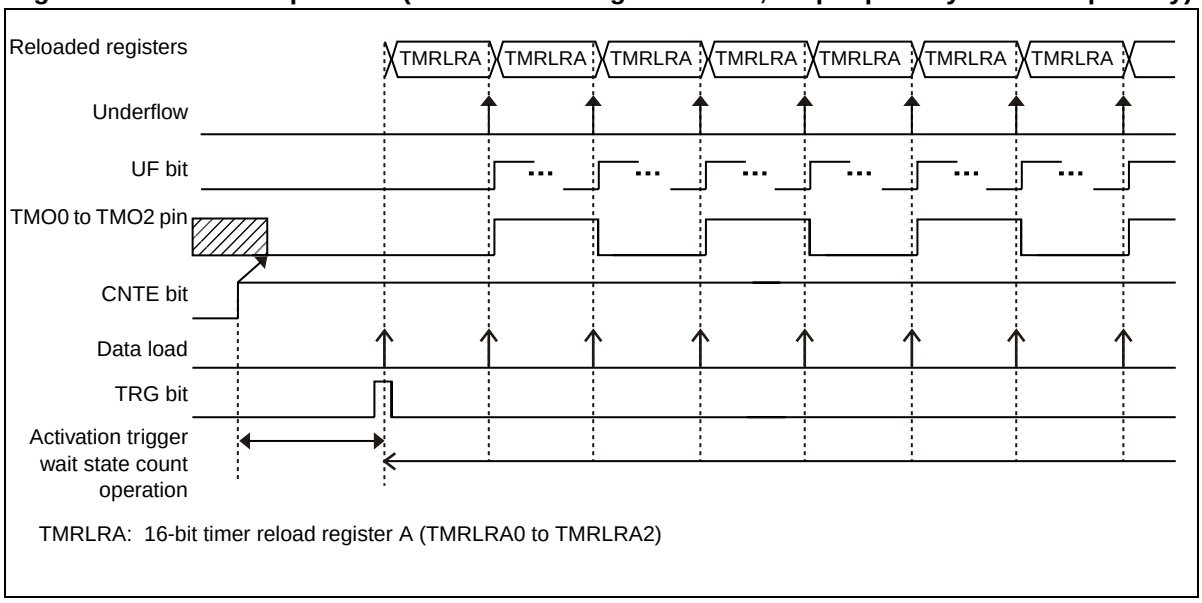
If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is inverted.
- The timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

As described, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

Figure 22.6-5 shows the count operation.

Figure 22.6-5 Count operation (activation through software, output polarity = normal polarity)



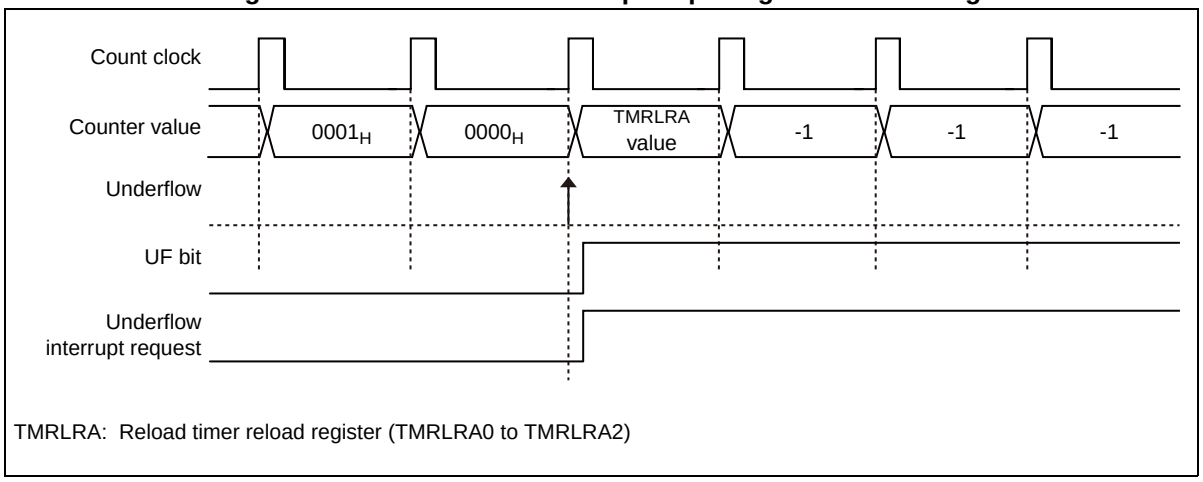
● Operation of interrupt processing

If the down counter enters an underflow condition , the UF bit of the timer control status register (TMCSR0 to TMCSR2) changes to "1".

In this case, if the INTE bit of the timer control status register (TMCSR0 to TMCSR2) is set to "1", an underflow interrupt request is generated.

Figure 22.6-6 shows the underflow interrupt request generation timing.

Figure 22.6-6 Underflow interrupt request generation timing



When "0" is written to the UF bit of the timer control status register (TMCSR0 to TMCSR2), the underflow interrupt request can be cleared.

<Note>

If an underflow interrupt request is generated at the same time the other underflow interrupt request is cleared, the clearing operation is ignored, and the underflow interrupt request remains generated.

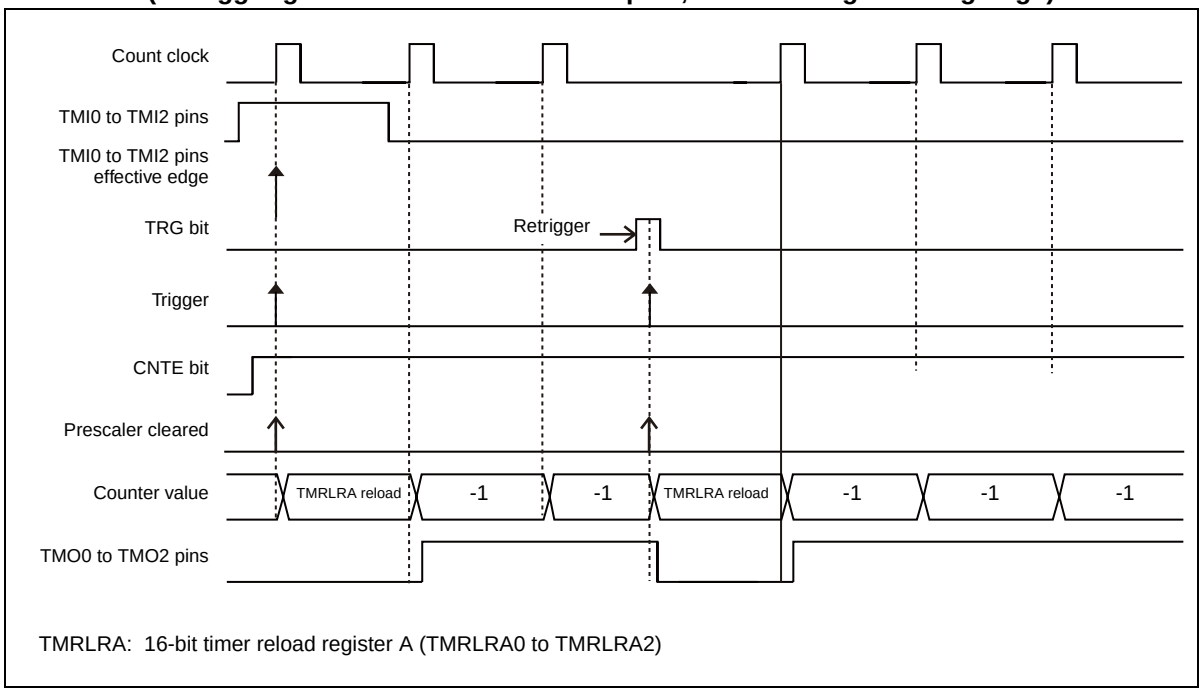
● Retrigger operation

If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

- The signal level of TMI0 to TMI2 pins is initialized.
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
- The prescaler is cleared.
- Count operation starts.

Figure 22.6-7 shows the operation when a retrigger is generated.

**Figure 22.6-7 Operation when a retrigger is generated.
(Retrigger generated on TMI0 to TMI2 pins, effective edge = rising edge)**



<Note>

When the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is rewritten, if a retrigger occurs at the same time the reload value is changed, the down counter loads the value before the change.

The value after change is loaded at the next reloading.

■ Operations in reload mode (TMI0 to TMI2 pins = at a gate input)

In this mode, TMI0 to TMI2 pins are used for gate input, and the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded each time underflow occurs to continue a countdown.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- TRGM0 bit = 0/1
- GATE bit = 1
- RELD bit = 1

● Activate

Use the following procedure for activating.

1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).

   The 16-bit reload timer enters the activation trigger wait state.

2. Input an activation trigger using the TRG bit of the timer control status register (TMCSR0 to TMCSR2). (TRG = 1)

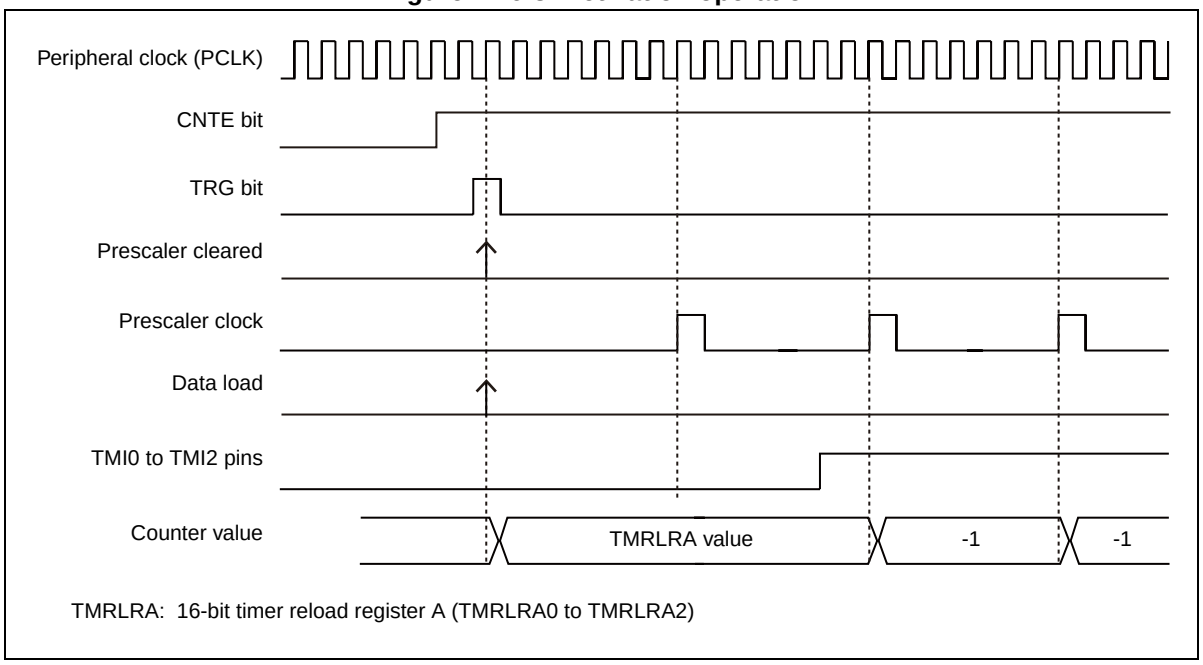
   The prescalar is cleared. The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and the 16-bit reload timer enters the effective input polarity (from TMI0 to TMI2 pins) wait state.

3. Input the signal with the level set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.

   The counter starts counting.

Figure 22.6-8 shows an activation operation.

Figure 22.6-8 Activation operation



<Note>

Be sure that the effective level input from TMI0 to TMI2 pins never falls below 2T (T: cycle of the peripheral clock (PCLK)).

● Count operation

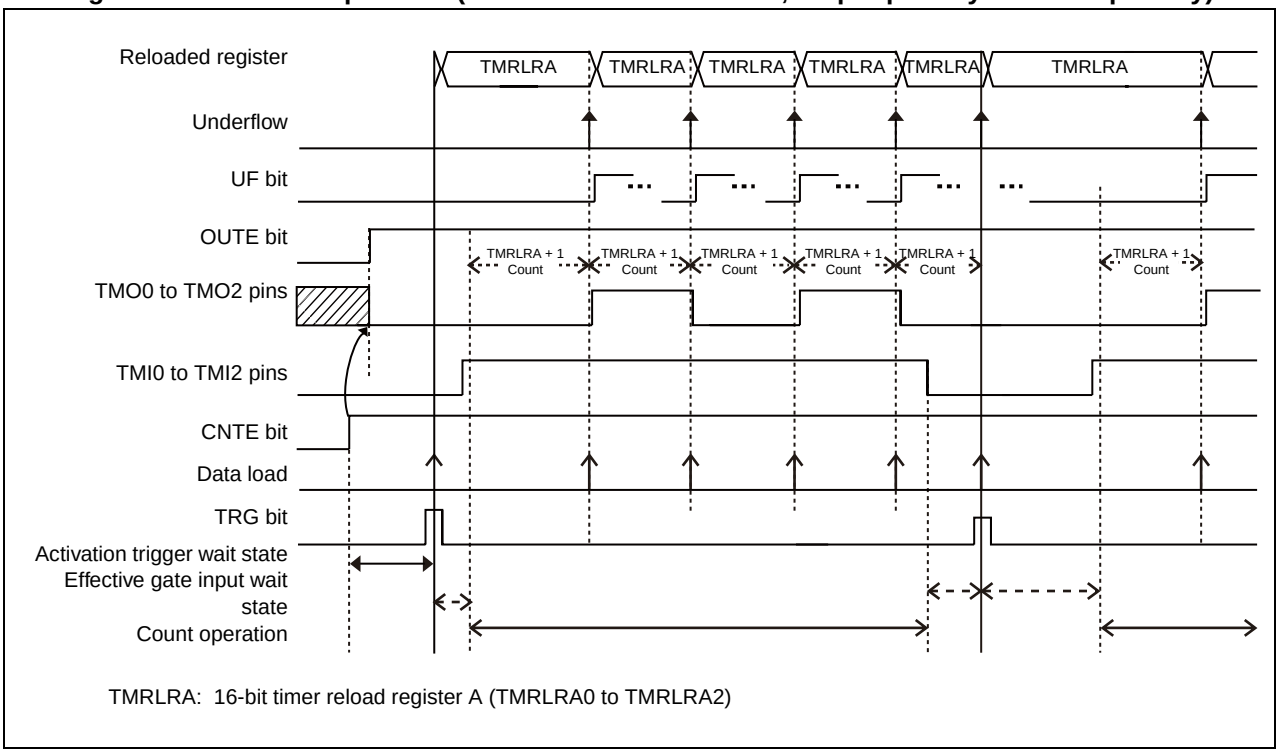
Only while the effective level signal is input from TMI0 to TMI2 pins does the down counter perform a countdown from the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) synchronously with the count clock.

If the effective level signal is not input from TMI0 to TMI2 pins, the down counter stops counting. If the effective level signal is input while the down counter is stopped, the counter starts counting from the value where it stopped.

The subsequent operations are the same as those when TMI0 to TMI2 pins = trigger input function is set. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

Figure 22.6-9 shows the count operation.

Figure 22.6-9 Count operation (effective level = "H" level, output polarity = normal polarity)



● Operation of interrupt processing

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

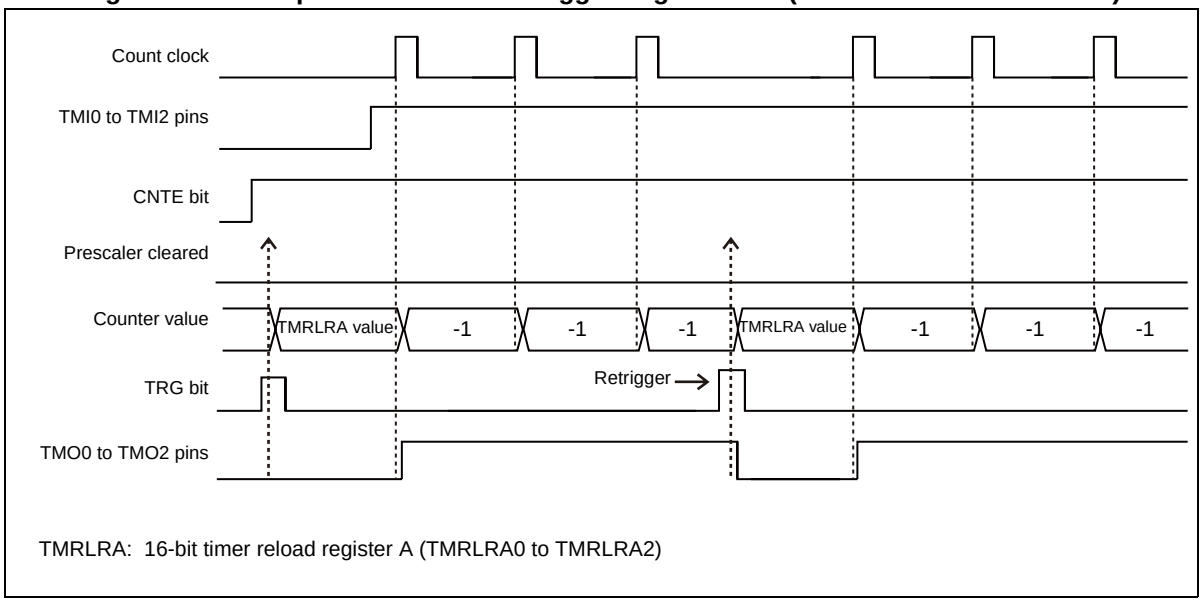
● Retrigger operation

If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

- The signal level of TMI0 to TMI2 pins is initialized.
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
- The prescaler is cleared.

When an effective level signal is input from TMI0 to TMI2 pin in such condition, counting starts. Figure 22.6-10 shows the operation when a retrigger is generated.

Figure 22.6-10 Operation when a retrigger is generated (effective level = "H" level)



■ Operations in one shot mode (TMI0 to TMI2 pins = trigger input)

When TMI0 to TMI2 pins are used for trigger input, if an underflow occurs, this mode stops counting until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- One of the TRGM1, TRGM0 bits = 01 to 11
- GATE bit = 0
- RELD bit = 0

● Activate

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

● Count operation

The down counter starts a countdown synchronously with the count clock from the value of 16-bit timer reload register A (TMRLRA0 to TMRLRA2).

If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is initialized.

- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

Figure 22.6-11 shows the count operation when TMI0 to TMI2 pins are used for activation.

Figure 22.6-11 Count operation (effective edge = rising edge, output polarity = normal polarity)

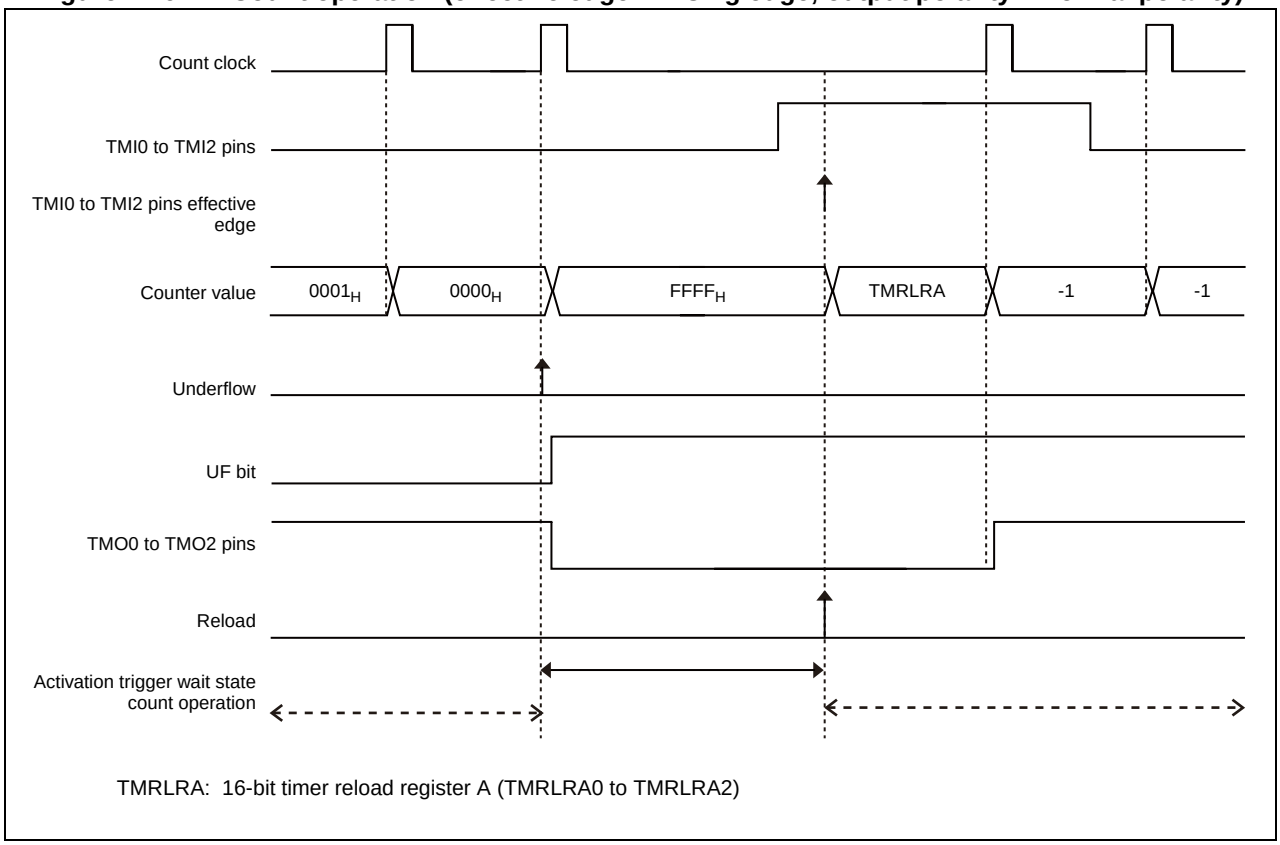
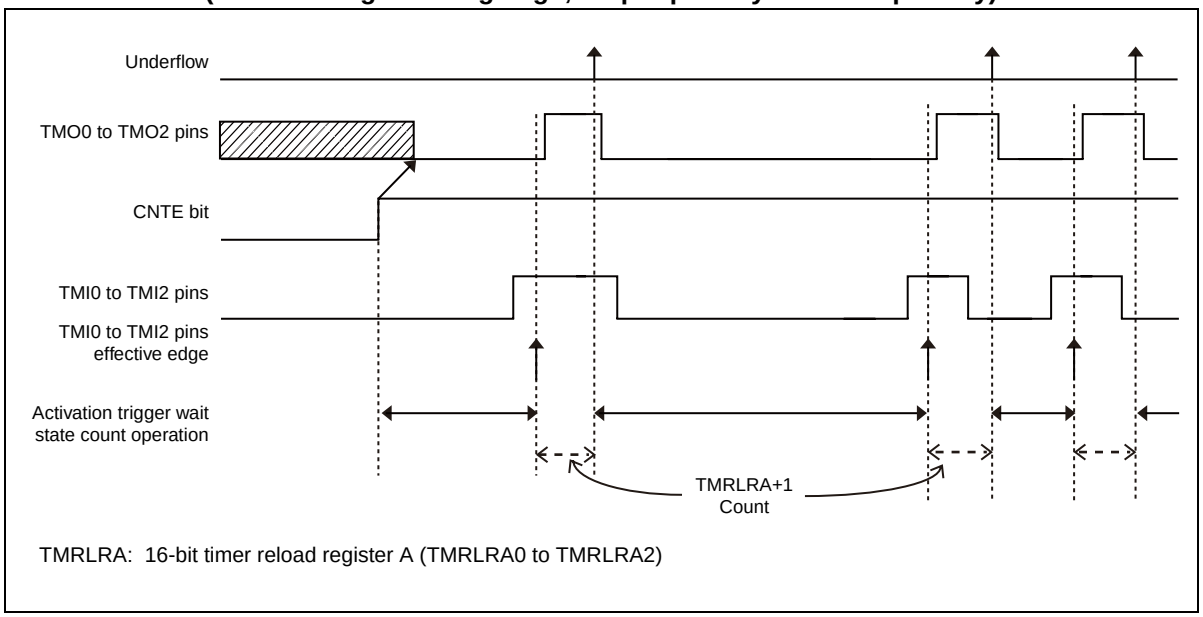


Figure 22.6-12 shows the detailed operation when an underflow occurs.

**Figure 22.6-12 Detailed operation when an underflow occurs.
(effective edge = rising edge, output polarity = normal polarity)**


Underflow
TMO0 to TMO2 pins
CNTE bit
TMI0 to TMI2 pins
TMI0 to TMI2 pins effective edge
Activation trigger wait state count operation
TMRLRA+1 Count
TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

● Operation of interrupt processing

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

● Retrigger operation

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

However, if a retrigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

■ Operations in one shot mode (TMI0 to TMI2 pins = gate input)

When TMI0 to TMI2 pins are used for gate input, if an underflow occurs, this mode stops counting until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- TRGM0 bit = 0/1
- GATE bit = 1
- RELD bit = 0

● Activate

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = at a gate input)".

However, if an activation trigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

● Count operation

Only while the effective level signal is input from TMI0 to TMI2 pins does the down counter counts down from the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) synchronously with count clock.

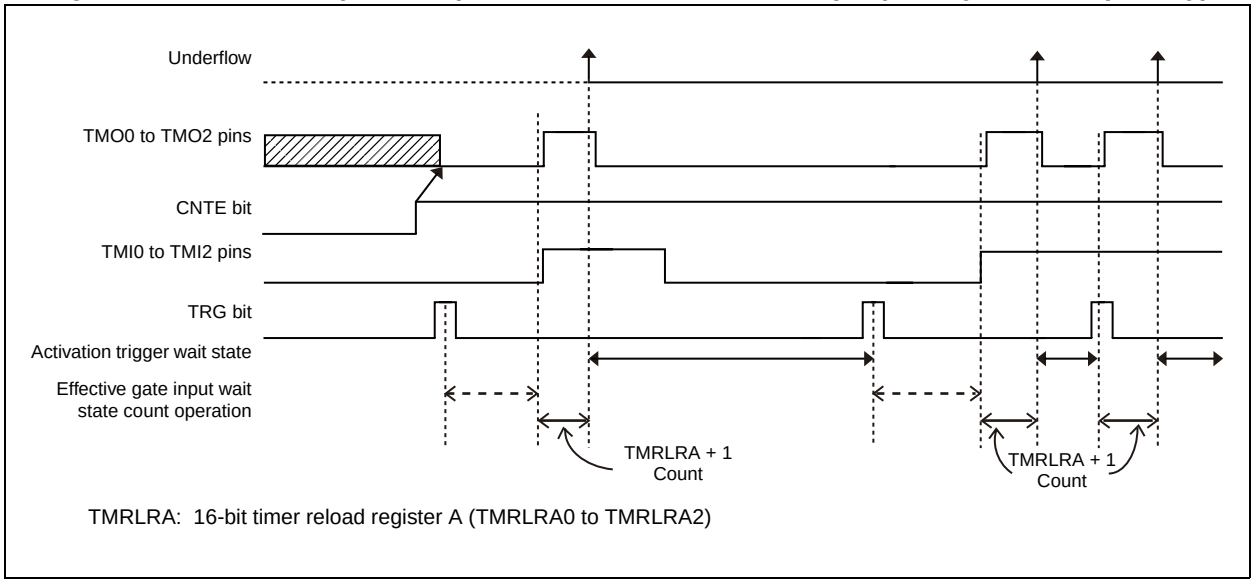
If the effective level signal is not input from TMI0 to TMI2 pins, the down counter stops counting. If the effective level signal is input while the down counter is stopped, the counter starts counting from the value where it stopped.

If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is initialized.
- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

Figure 22.6-13 shows the count operation.

Figure 22.6-13 Count operation (effective level = "H" level, output polarity = normal polarity)


Underflow
TMO0 to TMO2 pins
CNTE bit
TMI0 to TMI2 pins
TRG bit
Activation trigger wait state
Effective gate input wait state count operation
TMRLRA + 1 Count
TMRLRA + 1 Count
TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

● Operation of interrupt processing

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)".

● Retrigger operation

The operation is the same as in reload mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = at a gate input)".

However, if a retrigger is detected in one shot mode, the signal level of the signals output from TMO0 to TMO2 pins is inverted.

22.6.2 Operations in Event Counter Mode

This section explains the operations for using 16-bit reload timer as an event counter. This section explains the operation for counting external events.

■ Overview

In event counter mode, external events input from TMI0 to TMI2 pins are counted. It performs a countdown every time an effective edge is input from TMI0 to TMI2 pins.

For information on cascade mode, see "22.6.3 Operation in Cascade Mode".

■ Setting

● Event counter mode settings

To use the 16-bit reload timer in event counter mode, set CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2) as shown below.

CSL2	CSL1	CSL0	Mode	Count Clock
1	1	1	Event counter mode	External clock

● Operation mode settings

In event counter mode, one of the following operation modes can be selected using the RELD bits of the timer control status register (TMCSR0 to TMCSR2).

- Reload mode (RELD = 1)

  When the down counter enters an underflow condition, it reloads the value set to the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and repeats counting in this mode.
- One shot mode (RELD = 0)

  In this mode, counting stops when the down counter enters an underflow condition.

● Effective edge settings

The 16-bit reload timer performs a count down every time an effective edge is input from TMI0 to TMI2 pins.

The effective edge can be selected from the following settings of TRGM1 and TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2).

TRGM1, TRGM0	Pin Function
00	TMI0 to TMI2 pins do not work.
01	Rising edge
10	Falling edge
11	Both edges

■ Operation in reload mode

In this mode, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- One of the TRGM1, TRGM0 bits = 01 to 11
- RELD bit = 1

● Activate

Use the following procedure for activating.

1. Use the CNTE bit of the timer control status register (TMCSR0 to TMCSR2) to enable the operation of the 16-bit reload timer (CNTE = 1).

   The 16-bit reload timer enters the activation trigger wait state.

2. Input an activation trigger using the TRG bit of the timer control status register (TMCSR0 to TMCSR2). (TRG = 1)

   The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is loaded to the down counter, and the 16-bit reload timer enters the effective edge detection (of the signal output from TMI0 to TMI2 pins) wait state.

3. Input the effective edge set in the TRGM1, TRGM0 bits of the timer control status register (TMCSR0 to TMCSR2) from TMI0 to TMI2 pins.

   The counter starts counting.

● Count operation

Every time an effective edge is detected in the input signal from TMI0 to TMI2 pins, it performs a countdown.

Figure 22.6-14 to Figure 22.6-16 show the count timing.

Figure 22.6-14 Count timing (effective edge = rising edge)

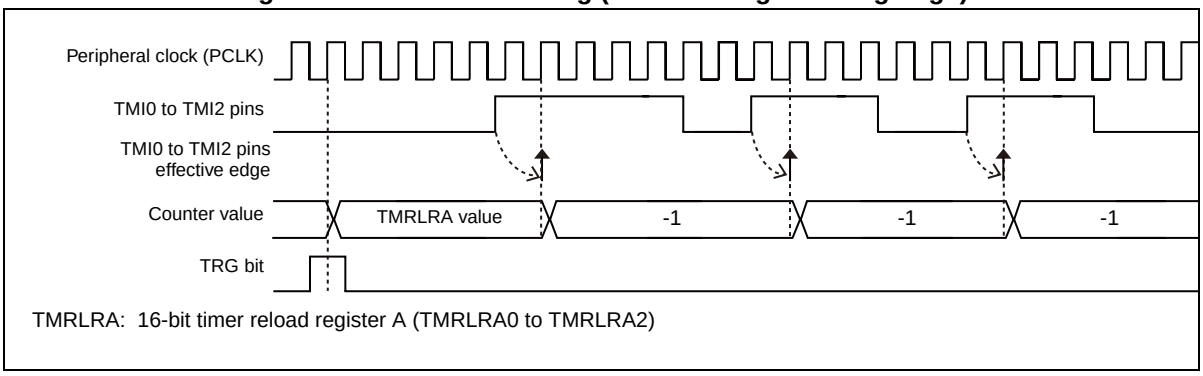


Figure 22.6-15 Count timing (effective edge = falling edge)

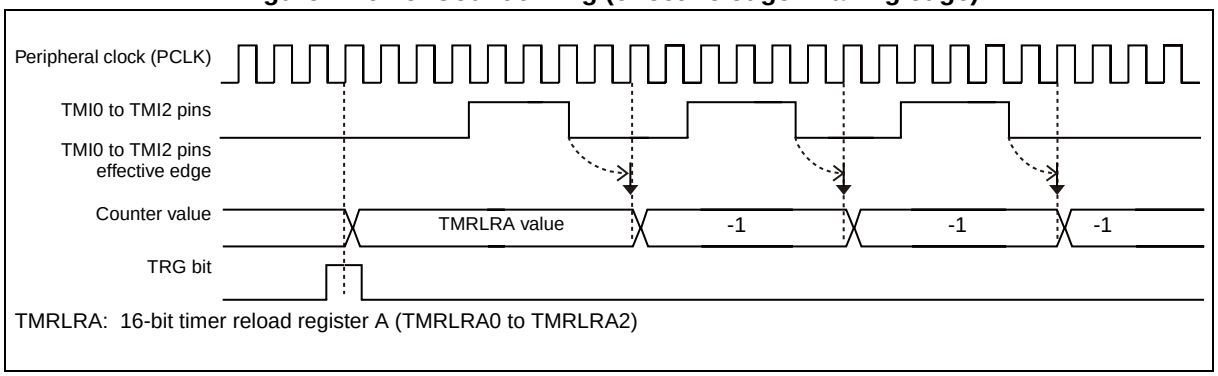
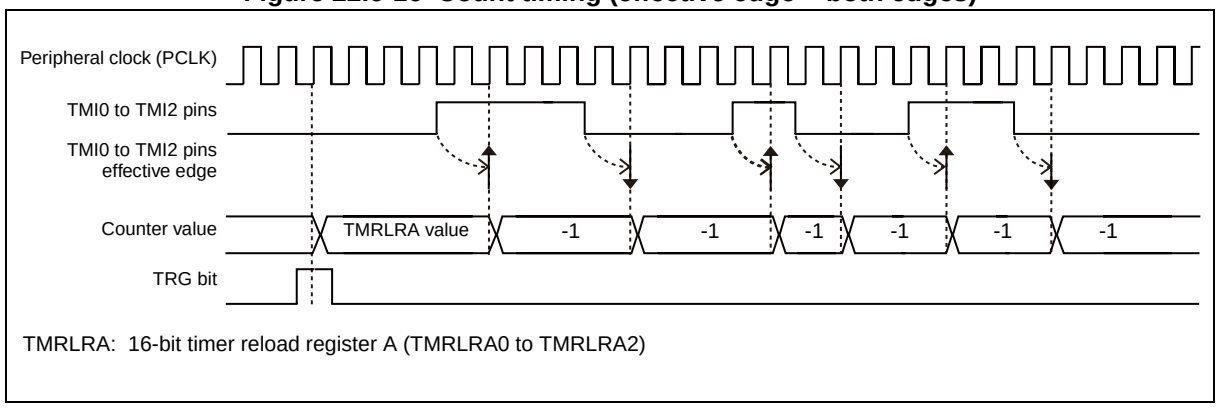


Figure 22.6-16 Count timing (effective edge = both edges)



If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

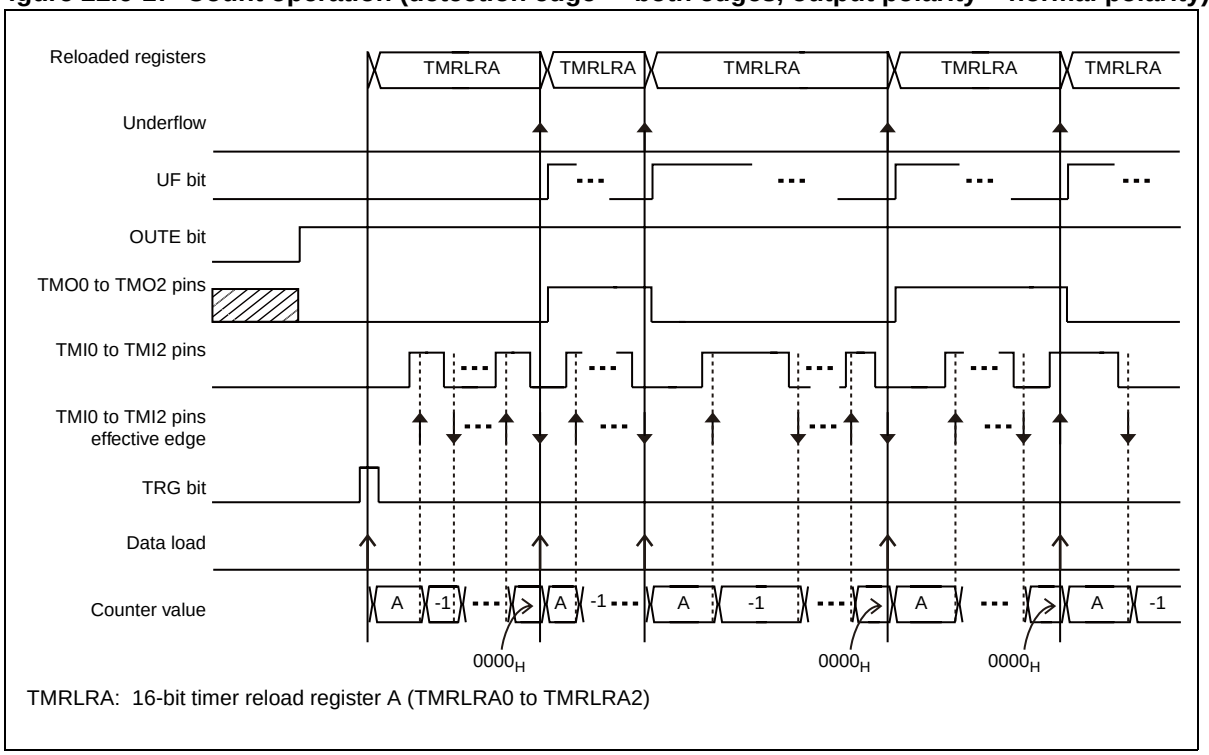
- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is inverted.
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.
- The counter continues counting when an effective level signal is input from TMI0 to TMI2 pins.

As described, every time an underflow occurs, the timer reloads the value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) and continues counting.

After an underflow occurs, counting does not start until an effective edge of the signal input from TMI0 to TMI2 pins is detected.

Figure 22.6-17 shows the count operation.

Figure 22.6-17 Count operation (detection edge = both edges, output polarity = normal polarity)



● Operation of interrupt processing

The operation is the same as in interval timer mode. See "■ Operations in reload mode (TMI0 to TMI2 pins = trigger input)" in "22.6.1 Operation in Interval Timer Mode".

● Retrigger operation

If an activation trigger of the 16-bit reload timer is detected during a count operation, the retrigger is generated, and the following operations are performed.

- The signal level of the signals output from TMO0 to TMO2 pins is initialized to the level set in the OUTL bit of the timer control status register (TMCSR0 to TMCSR2).
- The value of the 16-bit timer reload register A (TMRLRA0 to TMRLRA2) is reloaded to the down counter.

When an effective edge is input from TMI0 to TMI2 pin in such condition, counting starts.

■ Operation in one shot mode

When an underflow occurs, counting stops in this mode until the next activation trigger is input.

In this mode, set the timer control status register (TMCSR0 to TMCSR2) as follows:

- One of the TRGM1, TRGM0 bits = 01 to 11
- RELD bit = 0

● Activate

The operation is the same as in reload mode. See "■ Operation in reload mode".

● Count operation

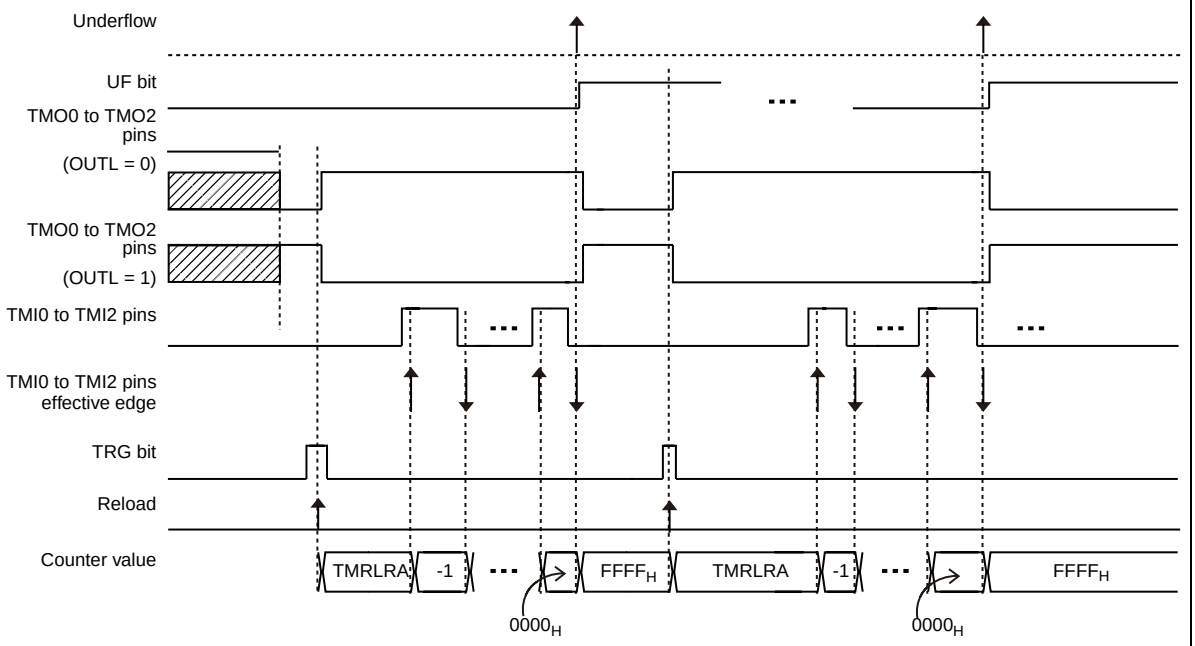
Every time an effective edge is detected from TMI0 to TMI2 pins, the counter counts down.

If counting starts from the down counter value "0000_H", an underflow occurs, and the following operations are performed.

- The UF bit of the timer control status register (TMCSR0 to TMCSR2) is changed to "1".
- The signal level of the signals output from TMO0 to TMO2 pins is initialized.
- Counting is stopped, and the counter enters the activation trigger wait state (the down counter value stops at "$FFFF_H$").

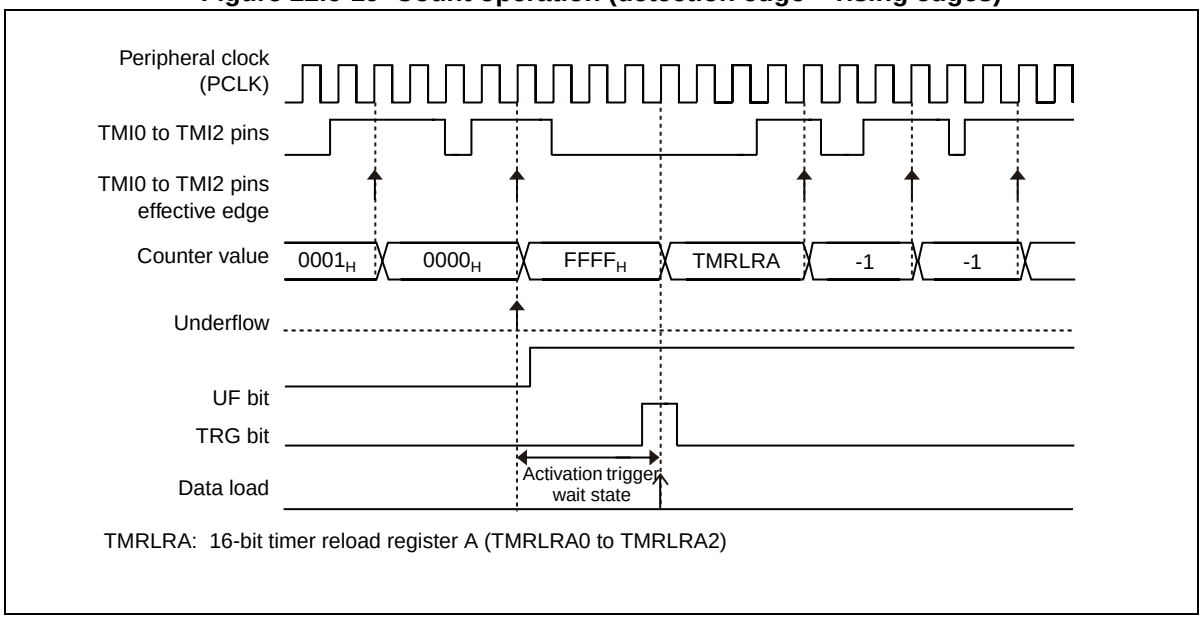
Figure 22.6-18 and Figure 22.6-19 show the count operations.

Figure 22.6-18 Count operation (detection edge = both edges)



TMRLRA: 16-bit timer reload register A (TMRLRA0 to TMRLRA2)

Figure 22.6-19 Count operation (detection edge = rising edges)



● Operation of interrupt processing

The operation is the same as in reload mode. See "■ Operation in reload mode".

● Retrigger operation

The operation is the same as in reload mode. See "■ Operation in reload mode".

22.6.3 Operation in Cascade Mode

In cascade mode, ch.1 can count the outputs from ch.0 of the 16-bit reload timer, and ch.2 can count the outputs from ch.1. This section explains the operations in cascade mode.

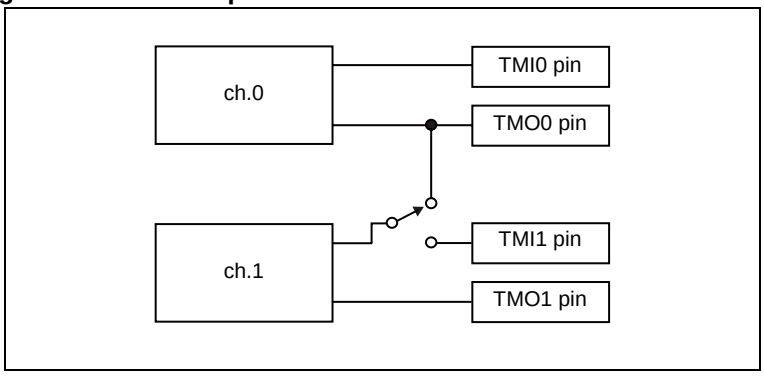
■ Operation

The following shows the count operation when cascade mode is selected with the CSL2 to CSL0 bits (CSL2 to CSL0 = 110) of the timer control status register (TMCSR0 to TMCSR2).

- When ch.1 is connected in cascade mode

  It counts the outputs from ch.0. Figure 22.6-20 shows the I/O operation when ch.1 is used in cascade mode.

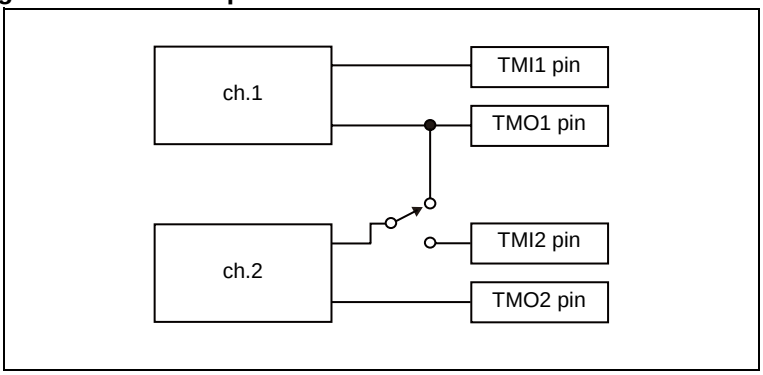
Figure 22.6-20 I/O operation when ch.1 is used in cascade mode



- When ch.2 is connected in cascade mode

  It counts the outputs from ch.1. Figure 22.6-21 shows the I/O operation when ch.2 is used in cascade mode.

Figure 22.6-21 I/O operation when ch.2 is used in cascade mode



<Note>

In cascade mode, use the CSL2 to CSL0 bits of the timer control status register (TMCSR0 to TMCSR2) to set the timer mode as shown below.

- Lower number channel

  Select interval timer mode or external clock (CSL2 to CSL0 = other than 110)

- Higher number channel

  Set cascade mode (CSL2 to CSL0 = 110)

■ Underflow cycle

This section explains the calculation of the underflow cycles of ch.1 and ch.2.

- When ch.1 is connected in cascade mode

T × (TMRLRA0 value + 1) × (TMRLRA1 value + 1)

T: Cycle of the count clock for ch.0
TMRLRA0: 16-bit timer reload register A0 (TMRLRA0)
TMRLRA1: 16-bit timer reload register A1 (TMRLRA1)

- When ch.2 is connected in cascade mode

T × (TMRLRA1 value + 1) × (TMRLRA2 value + 1)

T: Cycle of the count clock for ch.1
TMRLRA1: 16-bit timer reload register A1 (TMRLRA1)
TMRLRA2: 16-bit timer reload register A2 (TMRLRA2)

22.7 Notes on Use

Note the following points on using the 16-bit reload timer.

■ Notes on interrupts

- If an underflow interrupt request flag is cleared at the same time that it is set to "1", the clearing of the underflow interrupt request flag is ignored and the underflow interrupt request flag remains "1".

■ Operations for simultaneous activations

If more than one of the events used to determine the operating state of the 16-bit reload timer occur simultaneously, the priority order of these events is shown below.

1. Register reading
2. Trigger input
3. Underflow
4. Clock count input

CHAPTER 23 Base Timer I/O Select Function

This chapter explains the I/O select function of the base timer.

23.1 Overview
23.2 Configuration
23.3 Pin
23.4 Registers
23.5 I/O Mode

23.1 Overview

The I/O select function of the base timer determines the I/O method of the signals (external clock/ external activation trigger/wave form) to/from the base timer by setting the I/O mode.
In addition, the base timer can be used separately by channel as either of the following timers by switching the timer function.

- 16-bit PWM timer
- 16-bit PPG timer
- 16/32-bit reload timer
- 16/32-bit PWC timer

Be sure to use the base timer after reading both this chapter and the chapter on the timer function to be used.

■ Overview

The I/O mode can be selected from among the 9 types of modes for each 2 channels.

- I/O mode 0: 16-bit timer standard mode

  This mode operates the base timer individually, one channel at a time.

- I/O mode 1: Timer full mode

  In this mode, signals of the even-numbered channel of the base timer are allocated to the external pins separately to operate the timer.

- I/O mode 2: External trigger shared mode

  In this mode, the external activation trigger can be input to the 2 channels of base timers at the same time. This mode enables activating 2 channels of base timers at the same time.

- I/O mode 3: Other channel trigger shared mode

  In this mode, the external signal from other channels is input as an external activation trigger to activate the timer. This mode cannot be set for ch.0 and ch.1.

- I/O mode 4: Timer activation/stop mode

  This mode controls activation/stop of the odd-numbered channel by using the even-numbered channel. The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel and stops at the falling edge.

- I/O mode 5: Same time software activation mode

  This mode activates multiple channels at the same time using the software.

- I/O mode 6: Software activation timer activation/stop mode

  This mode controls activation/stop of the odd-numbered channel by using the even-numbered channel. The even-numbered channel is activated through software. The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel and stops at the falling edge.

- I/O mode 7: Timer activation mode

  This mode controls activation of the odd-numbered channel by using the even-numbered channel. The odd-numbered channel is activated at the rising edge of the output signal from the even-numbered channel.

- I/O mode 8: Other channel trigger shared timer activation/stop mode

  In this mode, the external signal from other channels is input as an external activation trigger to activate the timer. This mode cannot be set for ch.0 and ch.1.

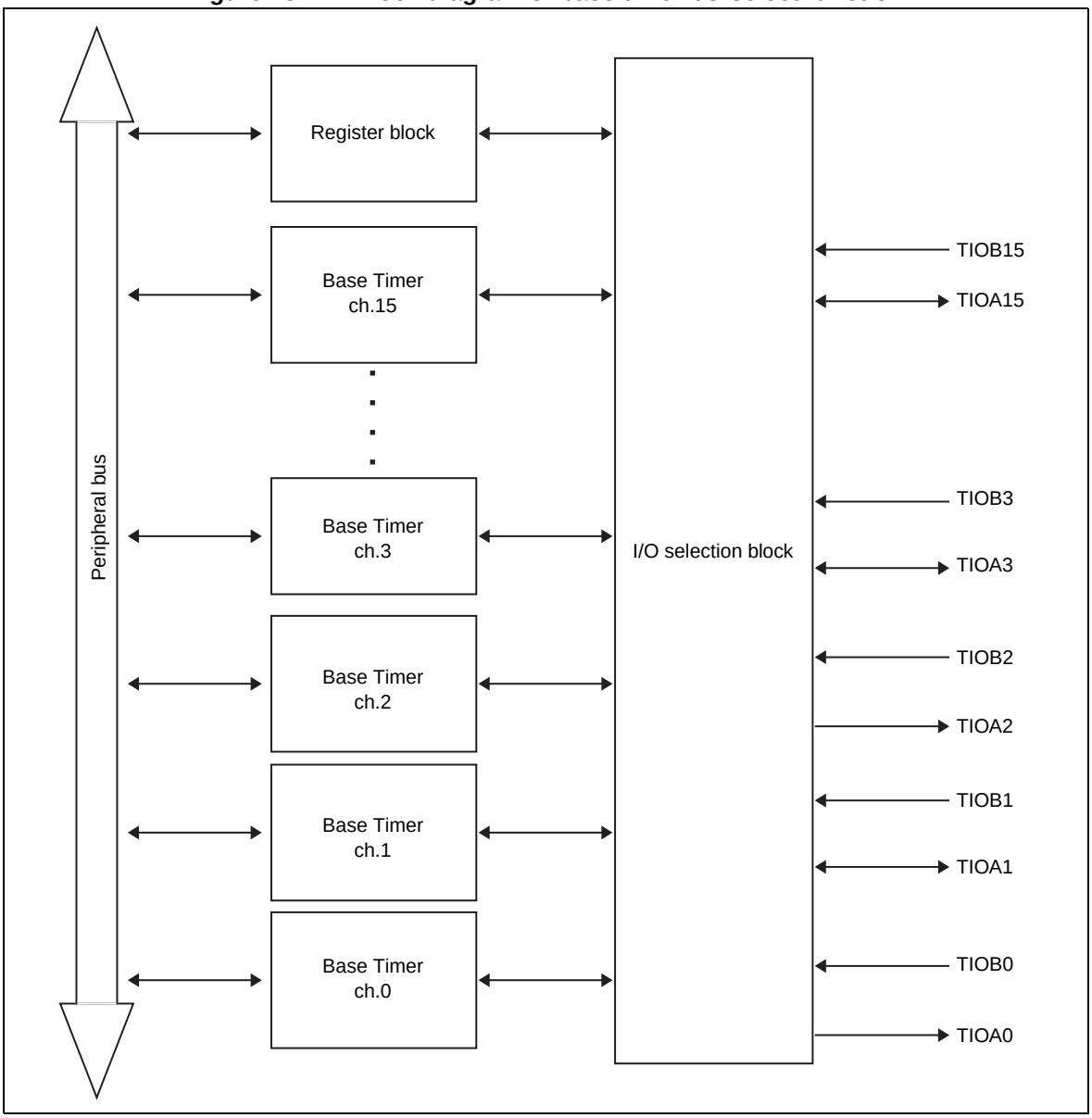
23.2 Configuration

The base timer I/O select function consists of the following blocks:

■ Block diagram of the base timer I/O select function

Figure 23.2-1 is a block diagram of the base timer I/O select function.

Figure 23.2-1 Block diagram of base timer I/O select function



- I/O selection block
  This circuit selects the I/O mode of the base timer for each channel.
- Base timer (ch.0 to ch.15)
  ch.0 to ch.15 of the base timer.

23.3 Pin

This section explains the pins for setting the I/O mode using the base timer I/O select function.

■ Overview

The base timer has 2 types of external pins and 5 types of internal signals for each channel.

By connecting the external pins and internal signals, signals that correspond to the connection destination (external clock (ECK signal)/external activation trigger (TGIN signal)/wave form (TIN signal)) are input to or output from the base timer.

The external pins and internal signals are connected by setting the I/O mode of the base timer. The pins that are used and the signals to be input/output vary depending on the I/O mode.

● External pin

- TIOA0 to TIOA15 pins

  These pins are used to output the wave form of the base timer (TOUT signal) or input the external activation trigger (TGIN signal).

  These pins are multiplexed pins. To use them as TIOA0 to TIOA15 pins of the base timer, see "2.4 Setting Method for Pins".

- TIOB0 to TIOB15 pins

  These pins are used to input the external activation trigger (TGIN signal)/external clock (ECK signal)/ wave form of another channel (TIN signal).

  These pins are multiplexed pins. To use them as TIOB0 to TIOB15 pins of the base timer, see "2.4 Setting Method for Pins".

● Internal signal

By connecting these pins to the above mentioned external pins or by inputting the output signal from another channel, signals is input to or output from the base timer.

- TOUT signal

  Output wave form of the base timer. (It is not used in the 16/32-bit PWC timer.)

- ECK signal

  External clock of the base timer. (It is not used in the 16/32-bit PWC timer.)

  This signal is input when the external clock is selected for the count clock.

- TGIN signal

  External activation trigger of the base timer. (It is not used in the 16/32-bit PWC timer.)

  When the effective edge of the external activation trigger is selected, the edge of this signal is detected to activate the base timer.

- TIN signal

  The wave form to be measured. (It is used only in the 16/32-bit PWC timer.)

- DTRG signal

  The base timer stops operation at the falling edge of this signal.

- COUT signal

  Output signal to other channels.

- CIN signal

  Signal that is input from other channels.

● Connection of the external pins and internal signals

The external pins and internal signals are connected by setting the I/O mode of the base timer. Table 23.3-1 outlines the relationship between the I/O mode and pin connections.

Table 23.3-1 Relationship between the I/O mode and pin connections

I/O Mode	TIOAn (Even-numbered Channel)		TIOBn (Even-numbered Channel)		TIOAn+1 (Odd-numbered Channel)		TIOBn+1 (Odd-numbered Channel)	
	Connection Destination	I/O	Connection Destination	I/O	Connection Destination	I/O	Connection Destination	I/O
0	ch.n's TOUT	Output	ch.n's ECK/ TGIN/TIN	Input	ch.n+1's TOUT	Output	ch.n+1's ECK/TGIN/ TIN	Input
1	ch.n's TOUT	Output	ch.n's ECK	Input	ch.n's TGIN	Input	ch.n's TIN	Input
2	ch.n's TOUT	Output	ch.n/ch.n+1's ECK/TGIN/ TIN*1	Input	ch.n+1's TOUT	Output	Not used	
3	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		
4	ch.n's TOUT	Output	ch.n's ECK/ TGIN/TIN	Input	ch.n+1's TOUT	Output		
5	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		
6	ch.n's TOUT	Output			ch.n+1's TOUT	Output		
7	ch.n's TOUT	Output	ch.n's ECK/ TGIN/TIN	Input	ch.n+1's TOUT	Output		
8	ch.n's TOUT	Output	Not used		ch.n+1's TOUT	Output		

ch.n even-numbered channel

ch.n+1 odd-numbered channel

n = 0, 2, 4, 6, 8, 10, 12, 14

*1 Synchronize with the peripheral clock (PCLK)

23.4 Registers

This section explains the configuration and functions of registers used in the base timer I/O select function.

■ List of registers of the base timer I/O select function

Table 23.4-1 lists registers of the base timer I/O select function.

Table 23.4-1 Registers of the base timer I/O select function

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	23.4.5
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	23.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	23.4.2
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	23.4.3
Common to 12 to 15	BTSELCDEF	Base timer io select register for ch.C/D/E/F	23.4.4

23.4.1 Base Timer IO Select Register for Ch.0/1/2/3 (BTSEL0123)

This register sets the I/O mode of ch.0 to ch.3 of the base timer.

Figure 23.4-1 shows the bit configuration of the base timer io select register for ch.0/1/2/3 (BTSEL0123).

Figure 23.4-1 Bit configuration of base timer io select register for ch.0/1/2/3 (BTSEL0123)

bit	7	6	5	4	3	2	1	0
	SEL23_3	SEL23_2	SEL23_1	SEL23_0	SEL01_3	SEL01_2	SEL01_1	SEL01_0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SEL23_3 to SEL23_0 (I/O select bit for ch.2/ch.3)

These bits set the I/O mode for ch.2 and ch.3 of the base timer.

SEL23_3	SEL23_2	SEL23_1	SEL23_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/ stop mode)

<Note>

Setting the values other than above is prohibited.

[bit3 to bit0]: SEL01_3 to SEL01_0 (I/O select bit for ch.0/ch.1)

These bits set the I/O mode of ch.0 and ch.1 of the base timer.

ch.0 and ch.1 are the lowest channels of the base timer so that modes that use signals from the lower side channels cannot be used in these channels. Therefore, the setting of the following modes is prohibited.

- I/O mode 3 (other channel trigger shared mode)
- I/O mode 8 (other channel trigger shared timer activation/stop mode)

SEL01_3	SEL01_2	SEL01_1	SEL01_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	Setting prohibited
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	Setting prohibited

<Note>

Setting the values other than above is prohibited.

23.4.2 Base Timer IO Select Register for Ch.4/5/6/7 (BTSEL4567)

This register sets the I/O mode of ch.4 to ch.7 of the base timer.

Figure 23.4-2 shows the bit configuration of the base timer io select register for ch.4/5/6/7 (BTSEL4567).

Figure 23.4-2 Bit configuration of base timer io select register for ch.4/5/6/7 (BTSEL4567)

bit	7	6	5	4	3	2	1	0
	SEL67_3	SEL67_2	SEL67_1	SEL67_0	SEL45_3	SEL45_2	SEL45_1	SEL45_0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SEL67_3 to SEL67_0 (I/O select bit for ch.6/ch.7)

These bits set the I/O mode of ch.6 and ch.7 of the base timer.

SEL67_3	SEL67_2	SEL67_1	SEL67_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/ stop mode)

<Note>

Setting the values other than above is prohibited.

[bit3 to bit0]: SEL45_3 to SEL45_0 (I/O select bit for ch.4/ch.5)

These bits set the I/O mode of ch.4 and ch.5 of the base timer.

SEL45_3	SEL45_2	SEL45_1	SEL45_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/ stop mode)

<Note>

Setting the values other than above is prohibited.

23.4.3 Base Timer IO Select Register for Ch.8/9/A/B (BTSEL89AB)

This register sets the I/O mode of ch.8 to ch.11 of the base timer.

Figure 23.4-3 shows the bit configuration of the base timer io select register for ch.8/9/A/B (BTSEL89AB).

Figure 23.4-3 Bit configuration of base timer io select register for ch.8/9/A/B (BTSEL89AB)

bit	7	6	5	4	3	2	1	0
	SELAB_3	SELAB_2	SELAB_1	SELAB_0	SEL89_3	SEL89_2	SEL89_1	SEL89_0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SELAB_3 to SELAB_0 (I/O select bit for ch.10/ch.11)

These bits set the I/O mode of ch.10 and ch.11 of the base timer.

SELAB_3	SELAB_2	SELAB_1	SELAB_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

<Note>

Setting the values other than above is prohibited.

[bit3 to bit0]: SEL89_3 to SEL89_0 (I/O select bit for ch.8/ch.9)

These bits set the I/O mode of ch.8 and ch.9 of the base timer.

SEL89_3	SEL89_2	SEL89_1	SEL89_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/ stop mode)

<Note>

Setting the values other than above is prohibited.

23.4.4 Base Timer IO Select Register for Ch.C/D/E/F (BTSELCDEF)

This register sets the I/O mode of ch.12 to ch.15 of the base timer.

Figure 23.4-4 shows the bit configuration of the base timer io select register for ch.C/D/E/F (BTSELCDEF).

Figure 23.4-4 Bit configuration of base timer io select register for ch.C/D/E/F (BTSELCDEF)

bit	7	6	5	4	3	2	1	0
	SELEF_3	SELEF_2	SELEF_1	SELEF_0	SELCD_3	SELCD_2	SELCD_1	SELCD_0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Change this register after setting the base timer to reset mode in FMD2 to FMD0 bits (FMD2 to FMD0 = 000) of the base timer x timer control register (BTxTMCR).

[bit7 to bit4]: SELEF_3 to SELEF_0 (I/O select bit for ch.14/ch.15)

These bits set the I/O mode of ch.14 and ch.15 of the base timer.

SELEF_3	SELEF_2	SELEF_1	SELEF_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

<Note>

Setting the values other than above is prohibited.

[bit3 to bit0]: SELCD_3 to SELCD_0 (I/O select bit for ch.12/ch.13)

These bits set the I/O mode of ch.12 and ch.13 of the base timer.

SELCD_3	SELCD_2	SELCD_1	SELCD_0	Explanation
0	0	0	0	I/O mode 0 (16-bit timer standard mode)
0	0	0	1	I/O mode 1 (timer full mode)
0	0	1	0	I/O mode 2 (external trigger shared mode)
0	0	1	1	I/O mode 3 (other channel trigger shared mode)
0	1	0	0	I/O mode 4 (timer activation/stop mode)
0	1	0	1	I/O mode 5 (same time software activation mode)
0	1	1	0	I/O mode 6 (software activation timer activation/stop mode)
0	1	1	1	I/O mode 7 (timer activation mode)
1	0	0	0	I/O mode 8 (other channel trigger shared timer activation/stop mode)

<Note>

Setting the values other than above is prohibited.

23.4.5 Base Timer Same Time Soft Start Register (BTSSSR)

This register simultaneously activates the base timers using the software.
Up to 16 channels corresponding to the bits in which "1" is written can be simultaneously activated.

Figure 23.4-5 shows the bit configuration of the base timer same time soft start register (BTSSSR).

Figure 23.4-5 Bit configuration of base timer same time soft start register (BTSSSR)

bit	15	14	13	12	11	10	9	8
	SSSR15	SSSR14	SSSR13	SSSR12	SSSR11	SSSR10	SSSR9	SSSR8
Attribute	W	W	W	W	W	W	W	W
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	SSSR7	SSSR6	SSSR5	SSSR4	SSSR3	SSSR2	SSSR1	SSSR0
Attribute	W	W	W	W	W	W	W	W
Initial value	X	X	X	X	X	X	X	X

W: Write only

X: Undefined

<Notes>

- Do not write to this register when the modes other than the following are set.
  - I/O mode 5 (same time software activation mode)
  - I/O mode 6 (software activation timer activation/stop mode) (only for even-numbered channels)
- For channels that are activated using this register, set the trigger input edge to the rising edge in the EGS1 and EGS0 bits (EGS1, EGS 0 = 01) of the base timer x timer control register (BTxTMCR).

[bit15]: SSSR15 (Same time software start bit for ch.15)

This bit activates the ch.15 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.15 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SELEF_3 to SELEF_0 bits of the base timer io select register for ch.C/D/E/F (BTSELCDEF) (SELEF_3 to SELEF_0 = 0101)

[bit14]: SSSR14 (Same time software start bit for ch.14)

This bit activates the ch.14 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.14 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SELEF_3 to SELEF_0 bits of the base timer io select register for ch.C/D/E/F (BTSELCDEF)
  - "5" (Same time software activation mode) (SELEF_3 to SELEF_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SELEF_3 to SELEF_0 = 0110)

[bit13]: SSSR13 (Same time software start bit for ch.13)

This bit activates the ch.13 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.13 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SELCD_3 to SELCD_0 bits of the base timer io select register for ch.C/D/E/F (BTSELCDEF) (SELCD_3 to SELCD_0 = 0101)

[bit12]: SSSR12 (Same time software start bit for ch.12)

This bit activates the ch.12 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.12 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SELCD_3 to SELCD_0 bits of the base timer io select register for ch.C/D/E/F (BTSELCDEF)
  - "5" (Same time software activation mode) (SELCD_3 to SELCD_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SELCD_3 to SELCD_0 = 0110)

[bit11]: SSSR11 (Same time software start bit for ch.11)

This bit activates the ch.11 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.11 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SELAB_3 to SELAB_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB) (SELAB_3 to SELAB_0 = 0101)

[bit10]: SSSR10 (Same time software start bit for ch.10)

This bit activates the ch.10 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.10 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SELAB_3 to SELAB_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB)
  - "5" (Same time software activation mode) (SELAB_3 to SELAB_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SELAB_3 to SELAB_0 = 0110)

[bit9]: SSSR9 (Same time software start bit for ch.9)

This bit activates the ch.9 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.9 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL89_3 to SEL89_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB) (SEL89_3 to SEL89_0 = 0101)

[bit8]: SSSR8 (Same time software start bit for ch.8)

This bit activates the ch.8 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.8 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL89_3 to SEL89_0 bits of the base timer io select register for ch.8/9/A/B (BTSEL89AB)
  - "5" (Same time software activation mode) (SEL89_3 to SEL89_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SEL89_3 to SEL89_0 = 0110)

[bit7]: SSSR7 (Same time software start bit for ch.7)

This bit activates the ch.7 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.7 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL67_3 to SEL67_0 bits of the base timer io select register for ch.4/5/6/7 (BTSEL4567) (SEL67_3 to SEL67_0 = 0101)

[bit6]: SSSR6 (Same time software start bit for ch.6)

This bit activates the ch.6 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.6 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL67_3 to SEL67_0 bits of the base timer io select register for ch.4/5/6/7 (BTSEL4567)
  - "5" (Same time software activation mode) (SEL67_3 to SEL67_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SEL67_3 to SEL67_0 = 0110)

[bit5]: SSSR5 (Same time software start bit for ch.5)

This bit activates the ch.5 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.5 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL45_3 to SEL45_0 bits of the base timer io select register for ch.4/5/6/7 (BTSEL4567) (SEL45_3 to SEL45_0 = 0101)

[bit4]: SSSR4 (Same time software start bit for ch.4)

This bit activates the ch.4 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.4 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL45_3 to SEL45_0 bits of the base timer io select register for ch.4/5/6/7 (BTSEL4567)
  - "5" (Same time software activation mode) (SEL45_3 to SEL45_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SEL45_3 to SEL45_0 = 0110)

[bit3]: SSSR3 (Same time software start bit for ch.3)

This bit activates the ch.3 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.3 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL23_3 to SEL23_0 bits of the base timer io select register for ch.0/1/2/3 (BTSEL0123) (SEL23_3 to SEL23_0 = 0101)

[bit2]: SSSR2 (Same time software start bit for ch.2)

This bit activates the ch.2 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.2 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL23_3 to SEL23_0 bits of the base timer io select register for ch.0/1/2/3 (BTSEL0123)
  - "5" (Same time software activation mode) (SEL23_3 to SEL23_0 = 0101)
  - "6" (Software activation timer activation/stop mode) (SEL23_3 to SEL23_0 = 0110)

[bit1]: SSSR1 (Same time software start bit for ch.1)

This bit activates the ch.1 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.1 of the base timer.*

* Only when the I/O mode is set to "5" (same time software activation mode) in SEL01_3 to SEL01_0 bits of the base timer io select register for ch.0/1/2/3 (BTSEL0123) (SEL01_3 to SEL01_0 = 0101)

[bit0]: SSSR0 (Same time software start bit for ch.0)

This bit activates the ch.0 of the base timer.

Written Value	Explanation
0	Ignored
1	Activates the ch.0 of the base timer.*

* Only when the I/O mode is set to either of the following modes in the SEL01_3 to SEL01_0 bits of the base timer io select register for ch.0/1/2/3 (BTSEL0123)
  - "5" (Same time software activation mode) SEL01_3 to SEL01_0)
  - "6" (Software activation timer activation/stop mode) (SEL01_3 to SEL01_0)

23.5 I/O Mode

Operations of the external pins and activation/stop timing of the base timer vary depending on the I/O mode set in the base timer io select register (BTSEL0123 to BTSELCDEF).

23.5.1 I/O Mode 0 (16-bit Timer Standard Mode)

In this mode, each channel of the base timer is used separately.

Table 23.5-1 lists the external pins used when this mode is set.

Table 23.5-1 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	1
Output pin	1	1

Table 23.5-2 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-2 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOA0 to TIOA15	Output	TOUT	Output wave form the base timer
TIOB0 to TIOB15	Input	ECK/TGIN/TIN*	Use the signals that have been input as one of the following: - External clock (ECK signal) - External activation trigger (TGIN signal) - Measured wave form (TIN signal)

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 23.5-1 is a block diagram of I/O mode 0 (16-bit timer standard mode), taking ch.0 as an example.

Figure 23.5-1 Block Diagram of I/O Mode 0 (16-bit Timer Standard Mode)

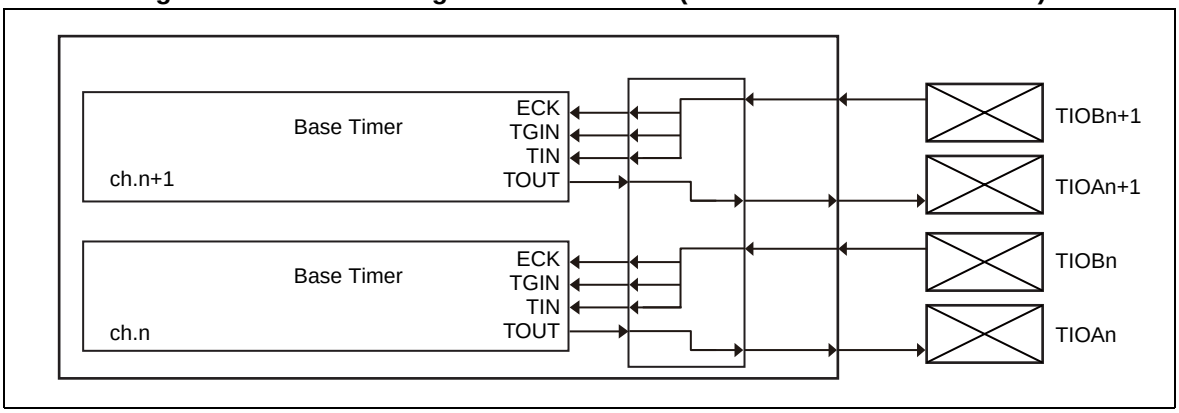


Table 23.5-3 lists the connections for I/O mode 0.

Table 23.5-3 Connections for I/O Mode 0

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to ch.n as TIN/TGIN/ECK
TOUT signal of ch.n+1	Output from the TIOAn+1 pin
Input signal from the TIOBn+1 pin	Input to ch.n+1 as TIN/TGIN/ECK

n=0, 2, 4, 6, 8, 10, 12, 14

23.5.2 I/O Mode 1 (Timer Full Mode)

In this mode, signals from the even-numbered channels are allocated to all the external pins separately to operate the timer.

Table 23.5-4 lists the external pins used when this mode is set.

Table 23.5-4 External Pins Used

Input pin	3
Output pin	1

Table 23.5-5 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-5 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOBn	Input	ECK of the even-numbered channel	Input the external clock (ECK signal) to the even-numbered channel
TIOAn+1	Input	TGIN of the even-numbered channel	Input the external activation trigger (TGIN signal) to the even-numbered channel
TIOBn+1	Input	TIN of the even-numbered channel	Input the measured wave form (TIN signal) in the even-numbered channel

n=0, 2, 4, 6, 8, 10, 12, 14

Figure 23.5-2 is a block diagram of I/O mode 1 (timer full mode).

Figure 23.5-2 Example of Block Diagram of I/O Mode 1 (Timer Full Mode)

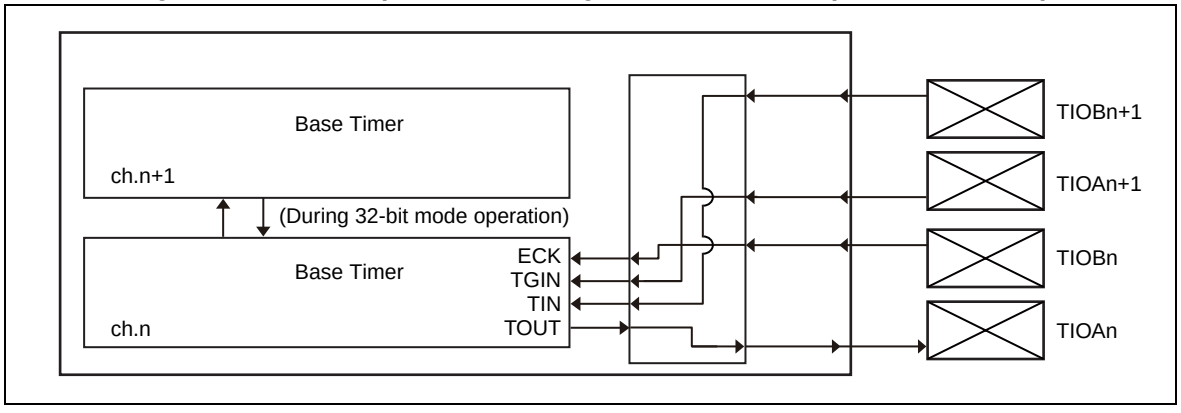


Table 23.5-6 lists the connections for I/O mode 1.

Table 23.5-6 Connections for I/O Mode 1

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Input signal from the TIOBn pin	Input to ch.n as a TIN signal
TOUT signal of ch.n+1	Input to ch.n as a TGIN signal
TIOBn+1 pin	Input to ch.n as an ECK signal

n=0, 2, 4, 6, 8, 10, 12, 14

<Note>

If this mode is set, set the TIOAn pins (TIOA1, TIOA3, TIOA5, ... TIOA15) corresponding to the odd-numbered channel to the port input mode in the port function register (PFR). For details of the setting of pins, see "2.4 Setting Method for Pins".

23.5.3 I/O Mode 2 (External Trigger Shared Mode)

In this mode, input signals to the base timer (ECK/TGIN/TIN) are shared by 2 channels.

Table 23.5-7 lists the external pins used when this mode is set.

Table 23.5-7 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1 (shared by 2 channels)	
Output pin	1	1

Table 23.5-8 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-8 Connection Destinations of the External Pins and I/O Signals

External pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even/odd-numbered channel*	Input to both of the even/odd-numbered channels (synchronized with the peripheral clock (PCLK)) and use it as one of the following: - External clock (ECK signal) - External activation trigger (TGIN signal) - Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10, 12, 14

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 23.5-3 is a block diagram of I/O mode 2 (external trigger shared mode).

Figure 23.5-3 Block Diagram of I/O Mode 2 (External Trigger Shared Mode)

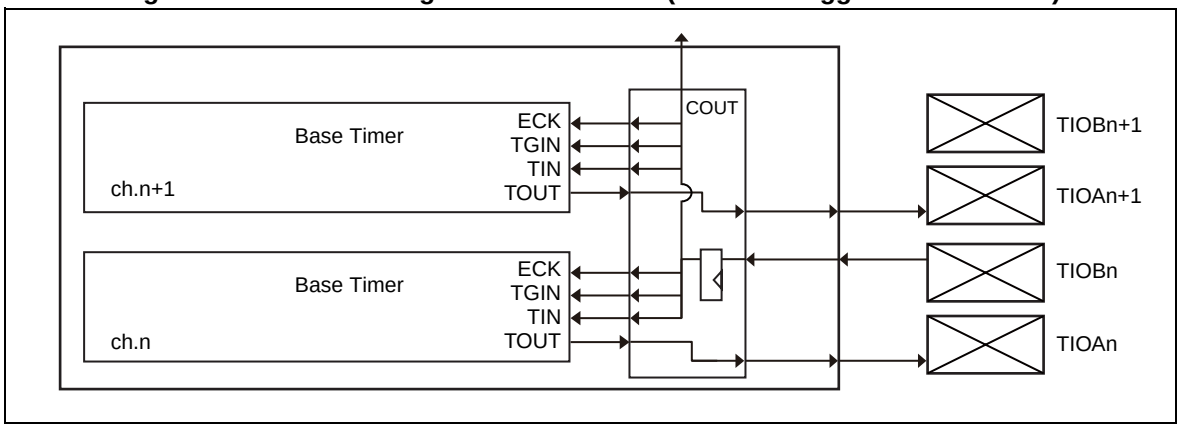


Table 23.5-9 lists the connections for I/O mode 2.

Table 23.5-9 Connections for I/O Mode 2

Connection Source	Connection Destination	Remarks
TOUT signal of ch.n	Output from the TIOAn pin	
Input signal from the TIOBn pin	- Input to ch.n and ch.n+1 as TIN/TGIN/ECK signals - Output to another channel as the COUT signal	Synchronization with the peripheral clock (PCLK)
TOUT signal of ch.n+1	Output from the TIOAn+1 pin	

n=0, 2, 4, 6, 8, 10, 12, 14

<Note>

If the upper 2 channels (n + 2, n + 3) of those that have been set to this mode are set to I/O mode 3 (other channel trigger shared mode), the input signals (ECK/TGIN/TIN) can be input to 4 channels at the same time.

(Example: If this mode is set for ch.0 and ch.1 and I/O mode 3 is set for ch.2 and ch.3, the input signals (ECK/TGIN/TIN) can be input to all 4 channels of ch.0 to ch.3 at the same time.)

23.5.4 I/O Mode 3 (Other Channel Trigger Shared Mode)

In this mode, the COUT signal of the channel that is lower by 2 channels is input as a CIN signal to be used as the ECK/TGIN/TIN signal.

Table 23.5-10 lists the external pins used when this mode is set.

Table 23.5-10 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 23.5-11 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-11 Connection Destinations of the External Pins and I/O Signals

External pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=2, 4, 6, 8, 10, 12, 14

Figure 23.5-4 is a block diagram of I/O mode 3 (other channel trigger shared mode).

Figure 23.5-4 Block Diagram of I/O Mode 3 (Other Channel Trigger Shared Mode)

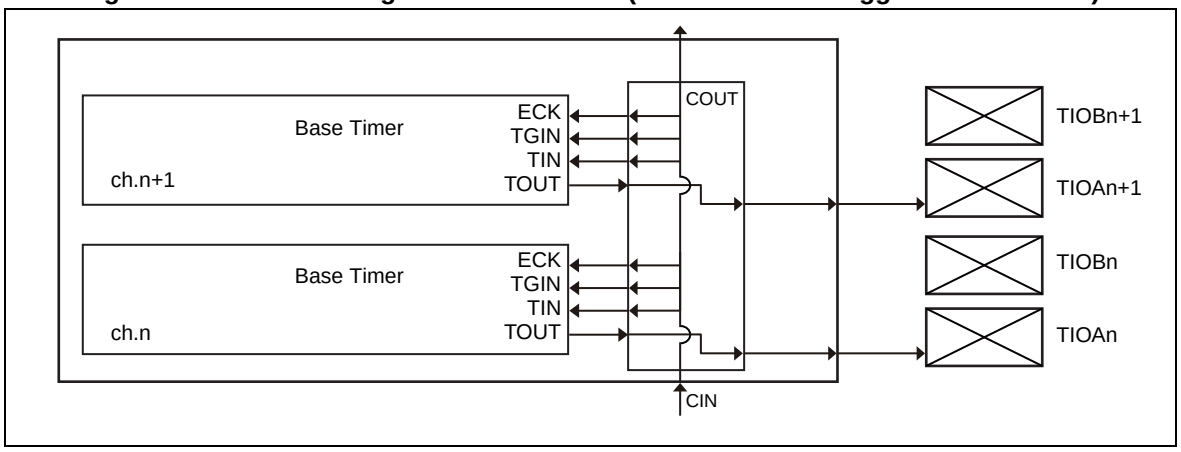


Table 23.5-12 lists the connections for I/O mode 3.

Table 23.5-12 Connections for I/O Mode 3

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal*	- Input to ch.n and ch.n+1 as the TIN/TGIN/ECK signal - Output to another channel as the COUT signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=2, 4, 6, 8, 10, 12, 14

* Input the COUT signal of the other channel as the CIN signal.

The signals of ch.n-2/n-1 that can be input to ECK, TGIN and TIN of ch.n/n+1 are as below.

- The signal that synchronized TIOBn-2 input of input/output mode 2 with peripheral clock.
- The trigger signal input from ch.n-4/n-3 of input/output mode 3.
- TIONAn-2 output of input/output mode 4.
- TIONAn-2 output of input/output mode 6.
- TIONAn-2 output of input/output mode 7.
- The trigger signal input from ch.n-4/n-3 of input/output mode 8.

<Notes>

- Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0=01) of the base timer x timer control register (BTxTMCR).
- Channels that have been set to this mode use the COUT signal of the channels (n - 2, n - 1) that are lower by 2 channels, as the CIN signal input.
  (Example: If ch.2 and ch.3 are set to this mode, they use the COUT signal of ch.0 and ch.1.)
  Therefore, ch.0 and ch.1 cannot be set to this mode.

23.5.5 Operations in I/O Mode 4 (Timer Activation/Stop Mode)

This mode enables control of activation/stop of the odd-numbered channel by using the even-numbered channel.
The odd-numbered channel is activated at the rising edge of the output wave form (TOUT signal) of the even-numbered channel and stops at the falling edge.

Table 23.5-13 lists the external pins used when this mode is set.

Table 23.5-13 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 23.5-14 lists the functions of pins.

Table 23.5-14 Functions of Pins

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even-numbered channel*	Input to the even-numbered channel and use as one of the following. - External clock (ECK signal) - External activation trigger (TGIN signal) - Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10, 12, 14

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 23.5-5 is a block diagram of I/O mode 4 (timer activation/stop mode).

Figure 23.5-5 Block Diagram of I/O Mode 4 (Timer Activation/Stop Mode)

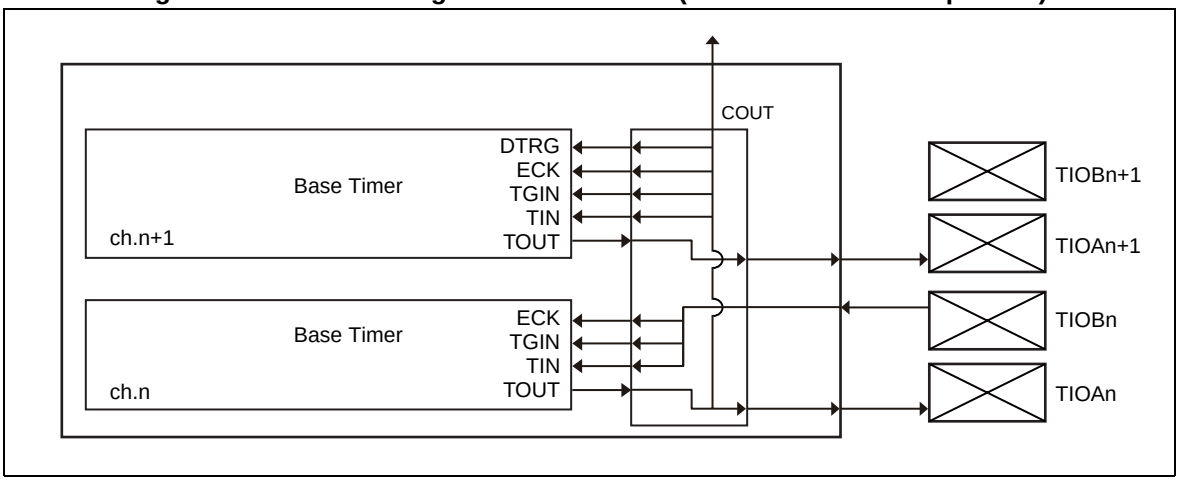


Table 23.5-15 lists the connections for I/O mode 4.

Table 23.5-15 Connections for I/O Mode 4

Connection Source	Connection Destination
TOUT signal of ch.n	- Output from the TIOAn pin - Input to ch.n+1 as the TIN/TGIN/ECK signal and DTRG signal - Output to another channel as the COUT signal
Input signal from the TIOBn pin	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6, 8, 10, 12, 14

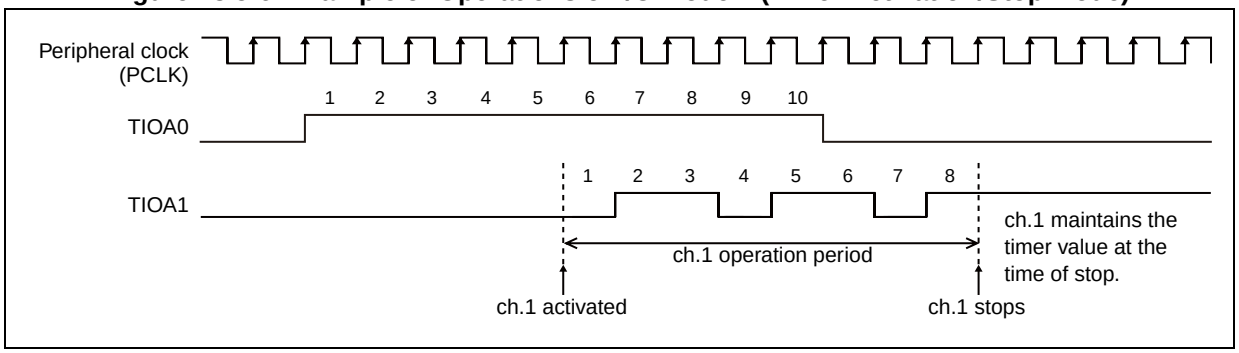
<Notes>

- Set the trigger input edge of the odd-numbered channel to the rising edge in the EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

Figure 23.5-6 shows the operation when I/O mode 4 (timer activation/stop mode) is set, taking as an example the case where ch.0 and ch.1 are used as the PWM timer.

Register (ch.0)	Setting Value	Register (ch.1)	Setting Value
Base timer 0 cycle setting register (BT0PCSR)	0010_H	Base timer 1 cycle setting register (BT1PCSR)	0002_H
Base timer 0 duty setting register (BT0PDUT)	0009_H	Base timer 1 duty setting register (BT1PDUT)	0001_H
Base timer 0 timer control register (BT0TMCR)	0013_H	Base timer 1 timer control register (BT1TMCR)	0112_H

Figure 23.5-6 Example of Operations of I/O Mode 4 (Timer Activation/Stop Mode)


Peripheral clock
(PCLK)
1 2 3 4 5 6 7 8 9 10
TIOA0
1 2 3 4 5 6 7 8
TIOA1
ch.1 maintains the timer value at the time of stop.
ch.1 operation period
ch.1 activated
ch.1 stops

23.5.6 Operations in I/O Mode 5 (Same Time Software Activation Mode)

This mode enables activating multiple channels at the same time by using the base timer same time soft start register (BTSSSR).

All channels corresponding to the bits in which "1" is written in the base timer same time soft start register (BTSSSR) are activated at the same time.

Table 23.5-16 lists the external pins used when this mode is set.

Table 23.5-16 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 23.5-17 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-17 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10, 12, 14

Figure 23.5-7 is a block diagram of I/O mode 5 (same time software activation mode).

Figure 23.5-7 Block Diagram of I/O Mode 5 (Same Time Software Activation Mode)

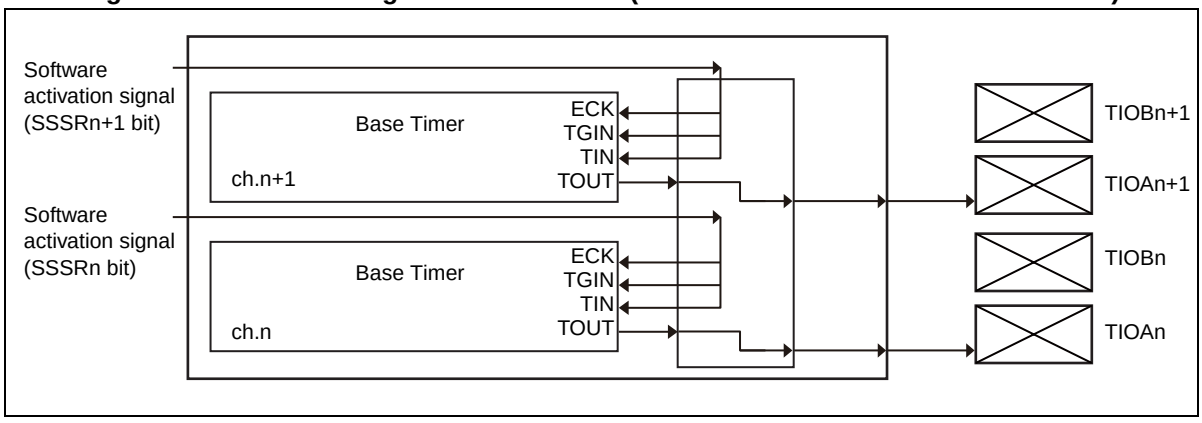


Table 23.5-18 lists the connections for I/O mode 5.

Table 23.5-18 Connections for I/O Mode 5

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
Software activation signal (Writing "1" in SSSRn bit of BTSSSR)	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin
Software activation signal (Writing "1" in SSSRn+1 bit of BTSSSR)	Input to ch.n+1 as the TIN/TGIN/ECK signal

n=0, 2, 4, 6, 8, 10, 12, 14

BTSSSR Base timer same time soft start register (BTSSSR)

If "1" is written in the base timer same time soft start register (BTSSSR), the rising edge is input (ECK/TGIN/TIN signal) in the channels that correspond to the written bits.

<Note>

Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).

23.5.7 Operations in I/O Mode 6 (Software Activation Timer Activation/Stop Mode)

This mode enables control of activation/stop of the odd-numbered channel by using the even-numbered channel.
The even-numbered channel is activated by writing "1" in the base timer same time soft start register (BTSSSR).
The odd-numbered channel is activated when the rising edge is detected in the output wave form (TOUT signal) of the even-numbered channel and stops when the falling edge is detected.

Table 23.5-19 lists the external pins used when this mode is set.

Table 23.5-19 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 23.5-20 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-20 Connection Destinations of the External Pins and I/O Signals

Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10, 12, 14

Figure 23.5-8 is a block diagram of I/O mode 6 (software activation timer activation/stop mode).

Figure 23.5-8 Block Diagram of I/O Mode 6 (Software Activation Timer Activation/Stop Mode)

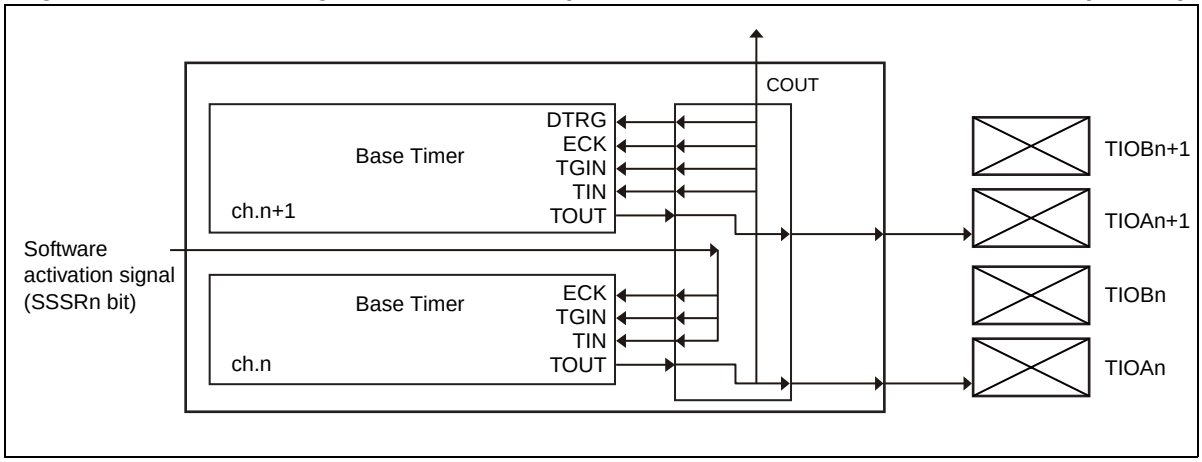


Table 23.5-21 lists the connections for I/O mode 6.

Table 23.5-21 Connections for I/O Mode 6

Connection Source	Connection Destination
TOUT signal of ch.n	- Output from the TIOAn pin - Input to ch.n+1 as the TIN/TGIN/ECK/DTRG signal - Output to another channel as the COUT signal
Software activation signal (Writing "1" in SSSRn bit of BTSSSR)	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6, 8, 10, 12, 14

BTSSSR Base timer same time soft start register (BTSSSR)

If "1" is written in the bits of the base timer same time soft start register (BTSSSR) that correspond to the even-numbered channels to be activated, the rising edge is input (ECK, TGIN, TIN signal) in the corresponding channels.

Start-up and stop timing of ch.n are same as input/output mode4.

<Notes>

- Set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

23.5.8 Operations in I/O Mode 7 (Timer Activation Mode)

In this mode, the output wave form (TOUT signal) of the even-numbered channel is used as input signals (ECK/TGIN/TIN signal) of the odd-numbered channel.

Table 23.5-22 lists the external pins used when this mode is set.

Table 23.5-22 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	1	Not used
Output pin	1	1

Table 23.5-23 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-23 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn	Input	ECK/TGIN/TIN of the even-numbered channel*	Input to the even-numbered channel and use as one of the following. - External clock (ECK signal) - External activation trigger (TGIN signal) - Measured wave form (TIN signal)
TIOBn+1	-	-	Not used

n=0, 2, 4, 6, 8, 10, 12, 14

* Input signals (ECK/TGIN/TIN signals) are used according to the base timer x timer control register (BTxTMCR) setting.

Figure 23.5-9 is a block diagram of I/O mode 7 (timer activation mode).

Figure 23.5-9 Block Diagram of I/O Mode 7 (Timer Activation Mode)

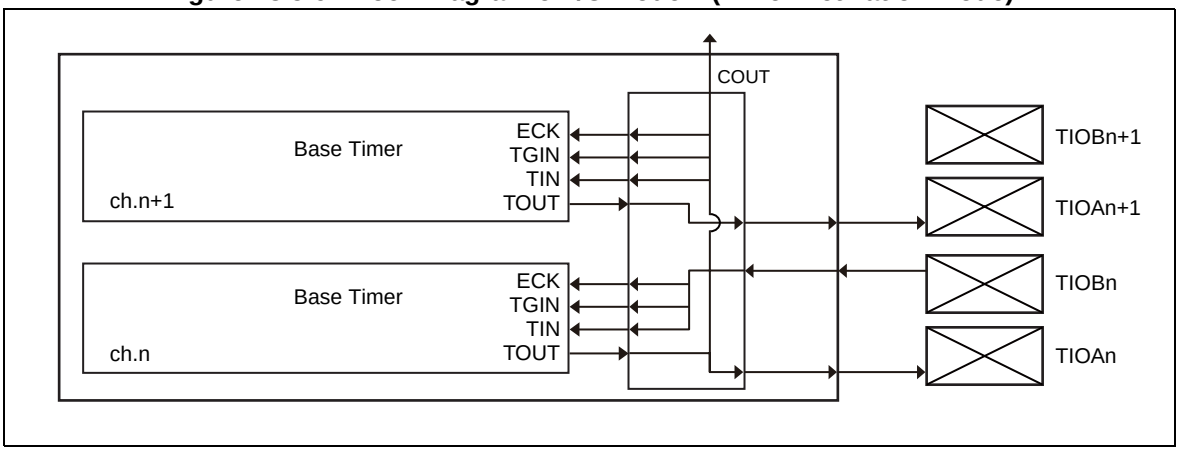


Table 23.5-24 lists the connection for I/O mode 7.

Table 23.5-24 Connection for I/O Mode 7

Connection Source	Connection Destination
TOUT signal of ch.n	- Output from the TIOAn pin - Input to ch.n+1 as the TIN/TGIN/ECK/DTRG signal - Output to another channel as the COUT signal
Input signal from the TIOBn pin	Input to ch.n as the TIN/TGIN/ECK signal
TOUT signal of ch.n+1	Output from the TIOAn+1 pin

n=0, 2, 4, 6, 8, 10, 12, 14

Start-up timing of ch.n is same as input/output mode4.

23.5.9 Operations in I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)

In this mode, the COUT signal of the channel that is lower by 2 channels is input as the CIN signal to be used as the external activation trigger (TGIN signal).

Table 23.5-25 lists the external pins used when this mode is set.

Table 23.5-25 External Pins Used

	Even-numbered Channel	Odd-numbered Channel
Input pin	Not used	
Output pin	1	1

Table 23.5-26 lists the connection destinations of the external pins used and I/O signals.

Table 23.5-26 Connection Destinations of the External Pins and I/O Signals

External Pin	I/O	Connection Destination (Internal Signal)	I/O Signal
TIOAn	Output	TOUT of an even-numbered channel	Output the wave form of an even-numbered channel
TIOAn+1	Output	TOUT of an odd-numbered channel	Output the wave form of an odd-numbered channel
TIOBn, TIOBn+1	-	-	Not used

n=2, 4, 6, 8, 10, 12, 14

Figure 23.5-10 is a block diagram of I/O mode 8 (other channel trigger shared timer activation/stop mode).

Figure 23.5-10 Block Diagram of I/O Mode 8 (Other Channel Trigger Shared Timer Activation/Stop Mode)

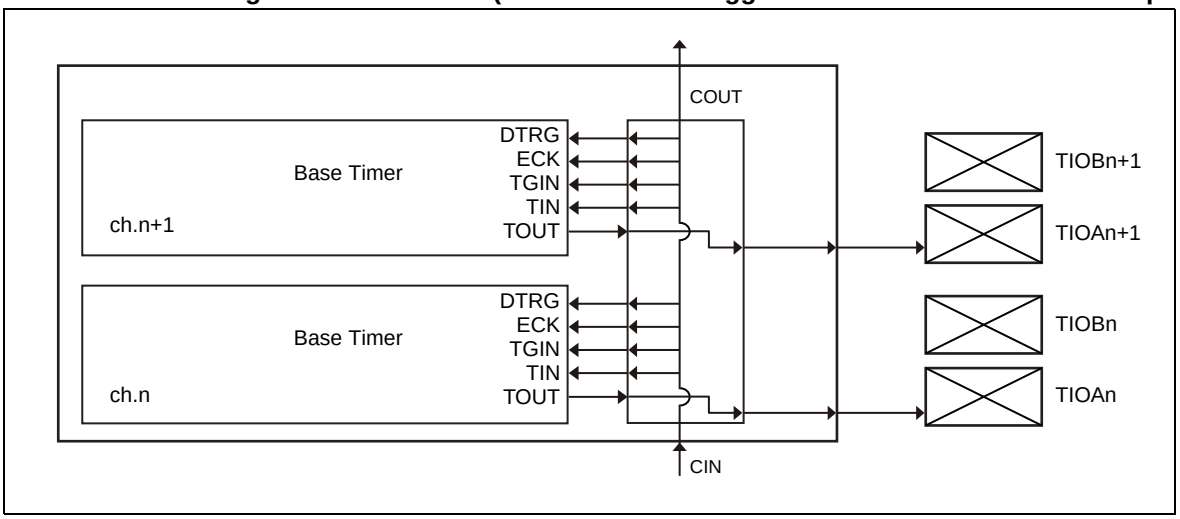


Table 23.5-27 lists the connections for I/O mode 8.

Table 23.5-27 Connections for I/O Mode 8

Connection Source	Connection Destination
TOUT signal of ch.n	Output from the TIOAn pin
CIN signal*	- Input to ch.n and ch.n+1 as the TIN/TGIN/ECK signal and DTRG signal - Output to another channel as the COUT signal

n=2, 4, 6, 8, 10, 12, 14

* Input the COUT signal of the other channel as the CIN signal.

The signals of ch.n-2/n-1 that can be input to ECK, TGIN and TIN of ch.n/n+1 are as below.

- The signal that synchronized TIOBn-2 input of input/output mode 2 with peripheral clock.
- The trigger signal input from ch.n-4/n-3 of input/output mode 3.
- TIONAn-2 output of input/output mode 4.
- TIONAn-2 output of input/output mode 6.
- TIONAn-2 output of input/output mode 7.
- The trigger signal input from ch.n-4/n-3 of input/output mode 8.

<Notes>

- Channels that have been set to this mode use the COUT signal of the channels (n - 2, n - 1) that are lower by 2 channels, as the CIN signal input.

  (Example: If ch.2 and ch.3 are set to this mode, they use the COUT signal of ch.0 and ch.1.)

  Therefore, ch.0 and ch.1 cannot be set to this mode.
- For the channels that are set to this mode, set the trigger input edge to the rising edge in EGS1 and EGS0 bits (EGS1, EGS0 = 01) of the base timer x timer control register (BTxTMCR).

  However, the above setting does not apply to the case where the timer function is set to 16/32-bit PWC timer in the FMD2 to FMD0 bits (FMD2 to FMD0 = 100) of the base timer x timer control register (BTxTMCR).
- The odd-numbered channel stops operation when the falling edge is detected in the DTRG signal.

CHAPTER 24 Base Timer

This chapter provides an overview of the base timer, summarizes its register configuration and functions, and describes its operations.

24.1 Overview of the Base Timer
24.2 Block Diagrams of the Base Timer
24.3 Base Timer's Registers
24.4 Operations of the Base Timer
24.5 32-bit Mode Operations
24.6 Notes of Using the Base Timer
24.7 Base Timer Interrupts
24.8 Base Timer Description by Function Mode

24.1 Overview of the Base Timer

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section outlines the base timer in each function mode available. This series is equipped with 16 channels.

■ Function Mode Bit Settings and Timer Function Modes Assigned

FMD2/FMD1/FMD0 bit Settings	Timer Function Mode
000_B	Reset mode
001_B	16-bit PWM timer
010_B	16-bit PPG timer
011_B	16/32-bit reload timer
100_B	16/32-bit PWC timer

■ Reset Mode

Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance.

■ 16-bit PWM Timer

The 16-bit PWM timer mainly consists of a 16-bit down counter, a 16-bit data register buffered for period setting, a 16-bit compare register buffered for duty cycle setting, and a pin controller.

Period data and duty cycle data can be updated during timer operation as they are held in their buffered respective registers.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The PWM timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PWM timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16-bit PPG Timer

The 16-bit PPG timer mainly consists of a 16-bit down counter, a 16-bit data register for "H"-width setting, a 16-bit data register for "L"-width setting, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The PPG timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the PPG timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16/32-bit Reload Timer

The 16/32-bit reload timer mainly consists of a 16-bit down counter, a 16-bit reload register, and a pin controller.

The count clock for the 16-bit down counter can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256) and three different external events (rising edge, falling edge and both edge detection).

The reload timer can select one-shot mode in which stops counting on an underflow or continuous mode in which repeats counting by reloading.

For activation, the reload timer can select a software trigger or one of three different external events (rising-edge detection, falling-edge detection, and both-edge detection).

■ 16/32-bit PWC Timer

The 16/32-bit PWC timer mainly consists of a 16-bit up counter, a measurement input pin, and control registers.

The PWC timer measures the time between arbitrary events based on the pulse input from an external source.

The reference count clock can be selected from among five different internal clocks (available by frequency-dividing the peripheral clock (PCLK) by 1, 4, 16, 128, and 256).

Measurement modes "H" pulse width (↑ to ↓) / "L" pulse width (↓ to ↑)
Rising period (↑ to ↑) / Falling period (↓ to ↓)
Inter-edge measurement (↑ or ↓ to ↓ or ↑)

The PWC timer can generate an interrupt request upon completion of measurement.

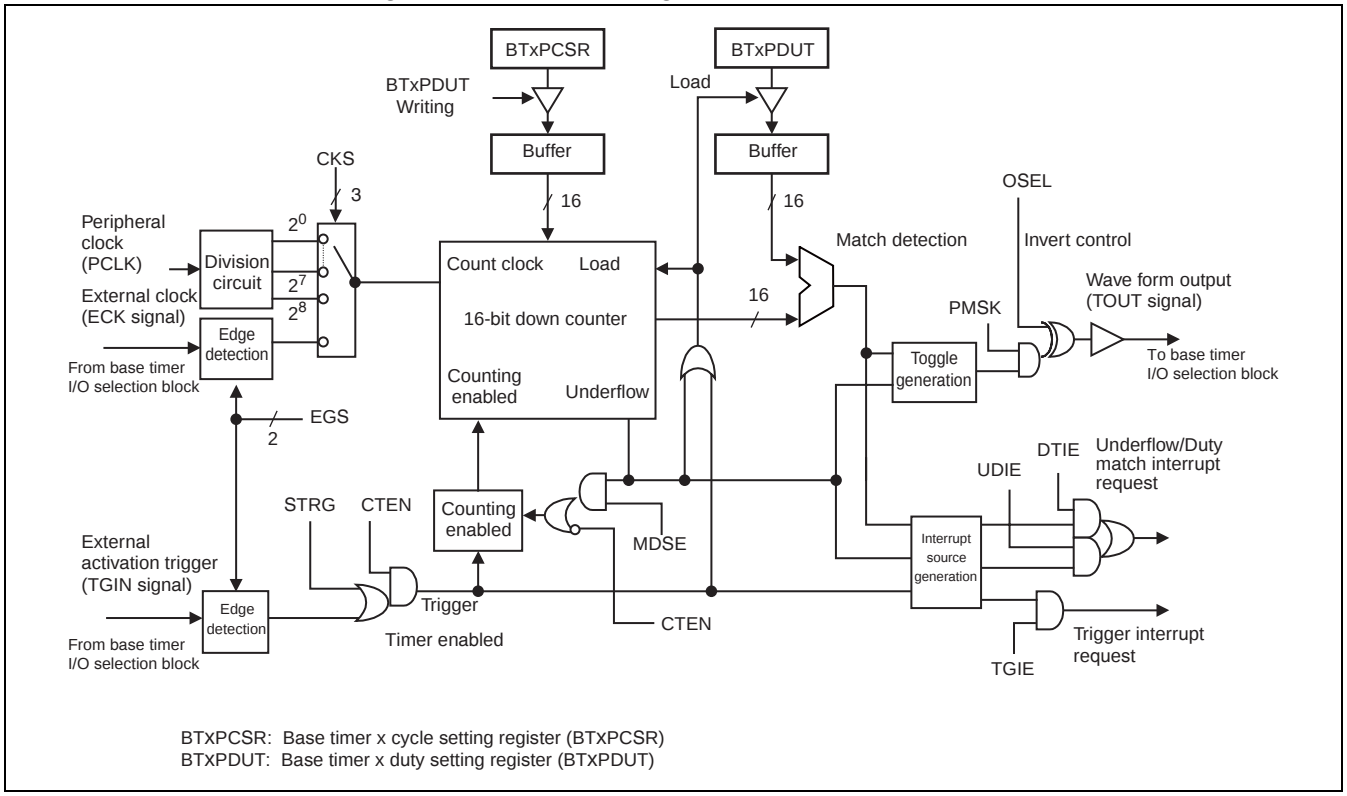
The PWC timer can select one-shot measurement or continuous measurement.

24.2 Block Diagrams of the Base Timer

This section provides a block diagram of the base timer in each function mode.

■ Block Diagram of 16-bit PWM Timer

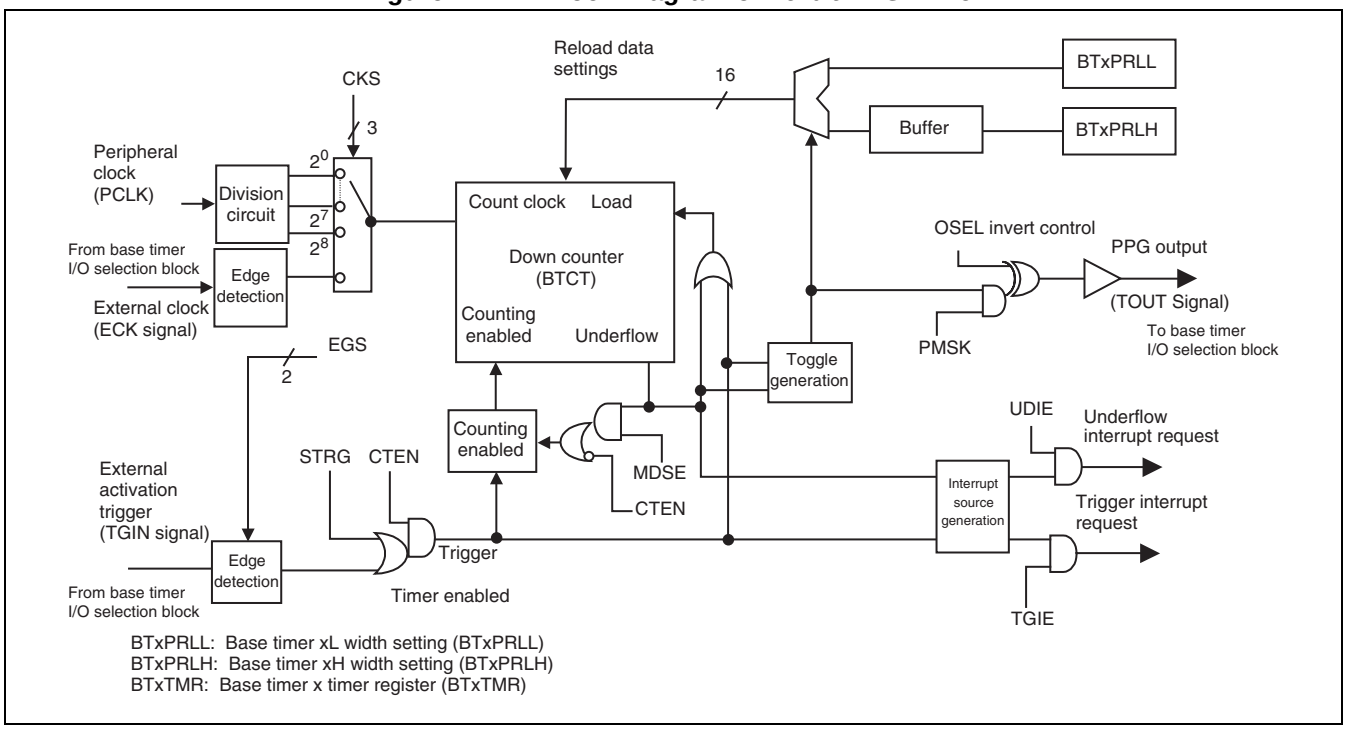
Figure 24.2-1 Block Diagram of 16-bit PWM Timer



BTxPCSR: Base timer x cycle setting register (BTxPCSR)
BTxPDUT: Base timer x duty setting register (BTxPDUT)

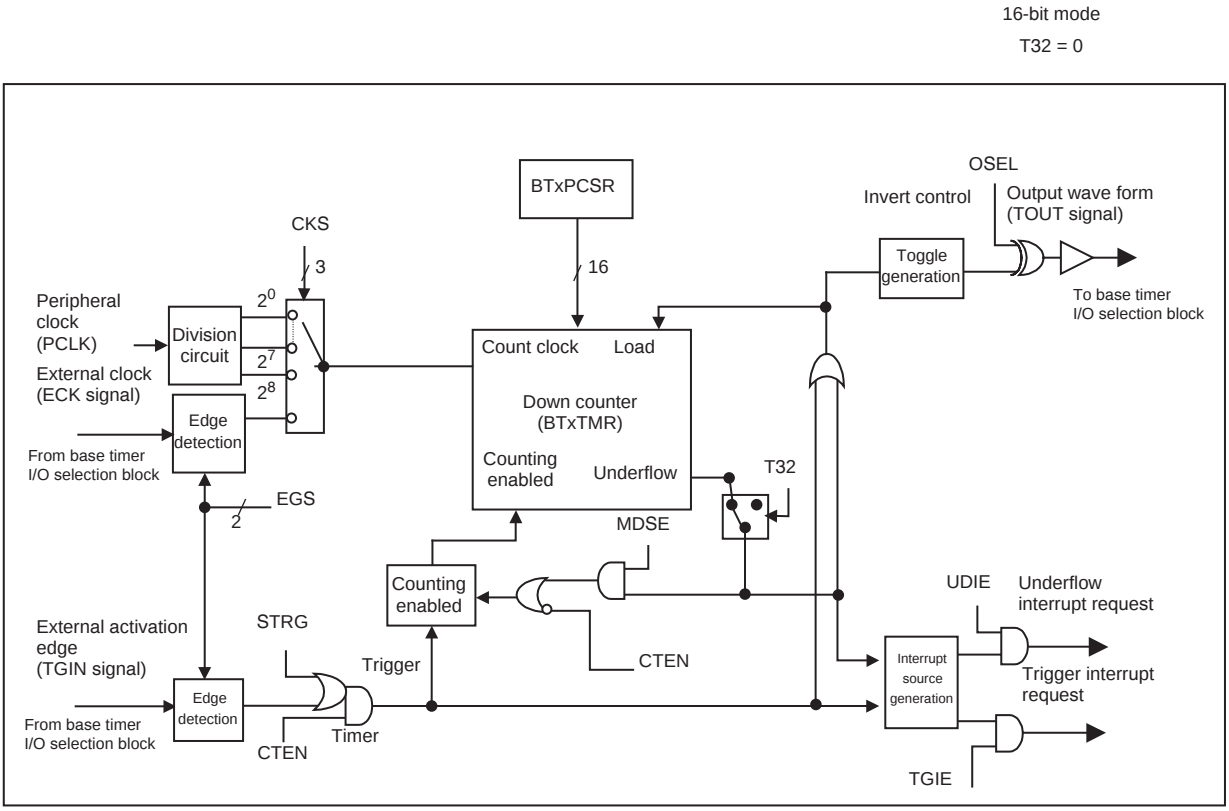
■ Block Diagram of 16-bit PPG Timer

Figure 24.2-2 Block Diagram of 16-bit PPG Timer



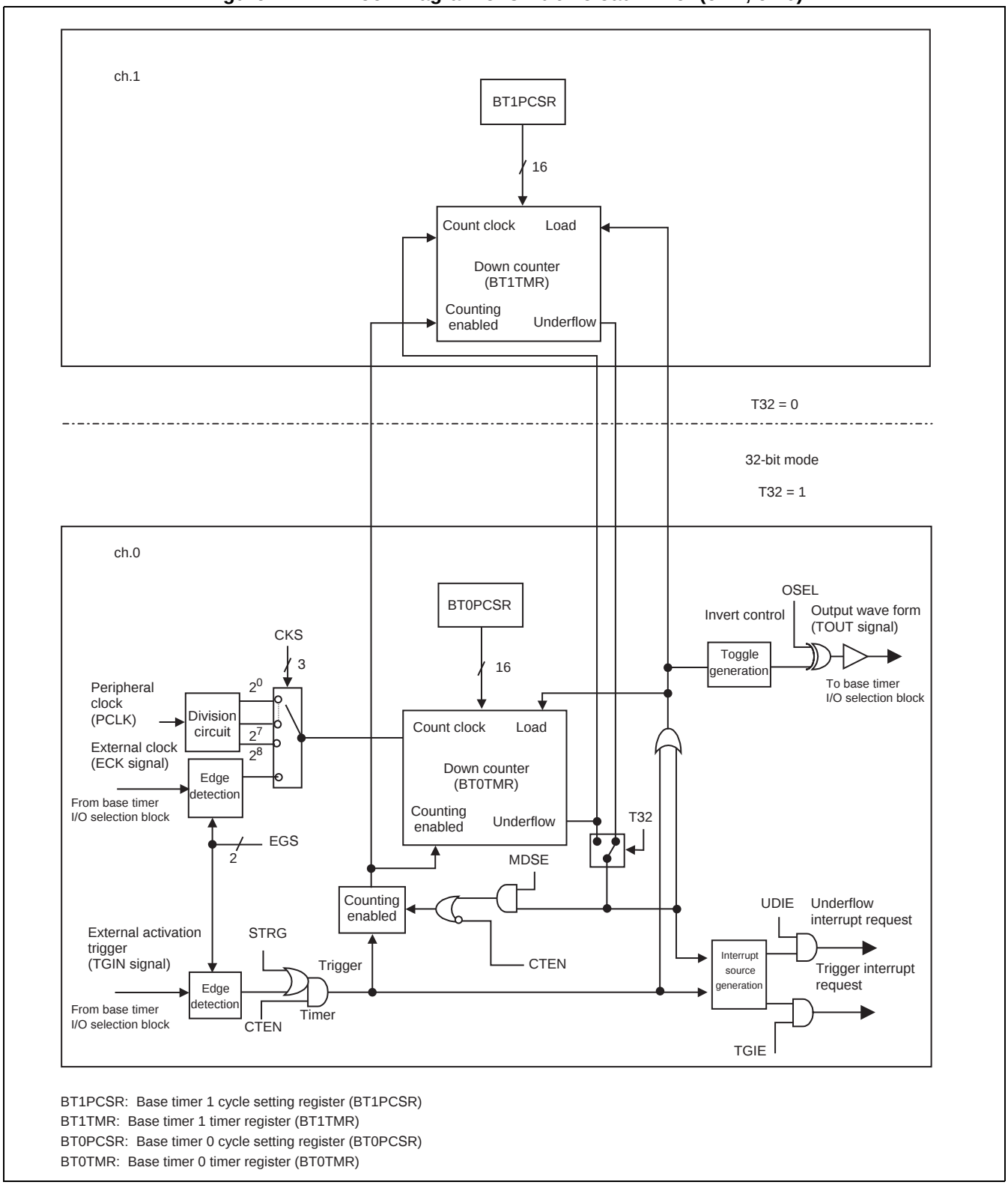
■ Block Diagram of 16/32-bit Reload Timer (ch.1, ch.0)

Figure 24.2-3 Block Diagram of 16-bit Reload Timer (ch.1, ch.0)



BTxPCSR: Base timer x cycle setting register (BTxPCSR)
BTxTMR: Base timer x timer register (BTxTMR)

Figure 24.2-4 Block Diagram of 32-bit Reload Timer (ch.1, ch.0)



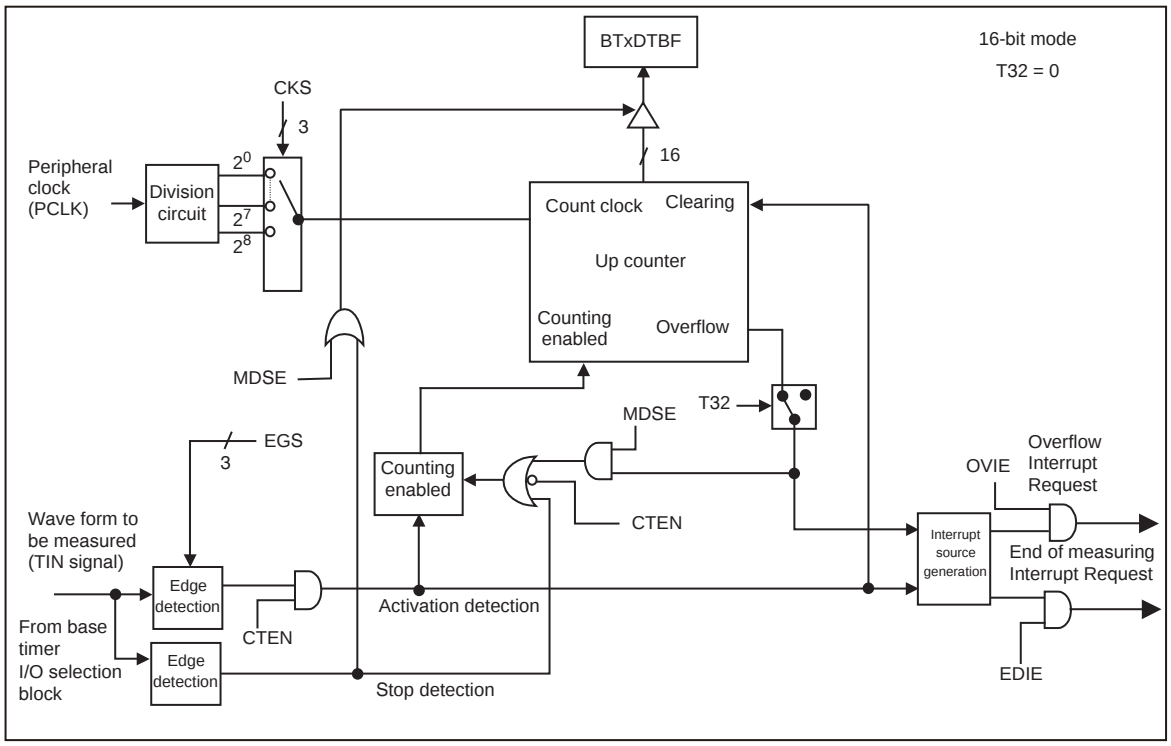
BT1PCSR: Base timer 1 cycle setting register (BT1PCSR)
BT1TMR: Base timer 1 timer register (BT1TMR)
BT0PCSR: Base timer 0 cycle setting register (BT0PCSR)
BT0TMR: Base timer 0 timer register (BT0TMR)

<Notes>

- The reload timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, between ch.6 and ch.7, between ch.8 and ch.9, between ch.10 and ch.11, between ch.12 and ch.13, and between ch.14 and ch.15. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 23 Base Timer I/O Select Function".

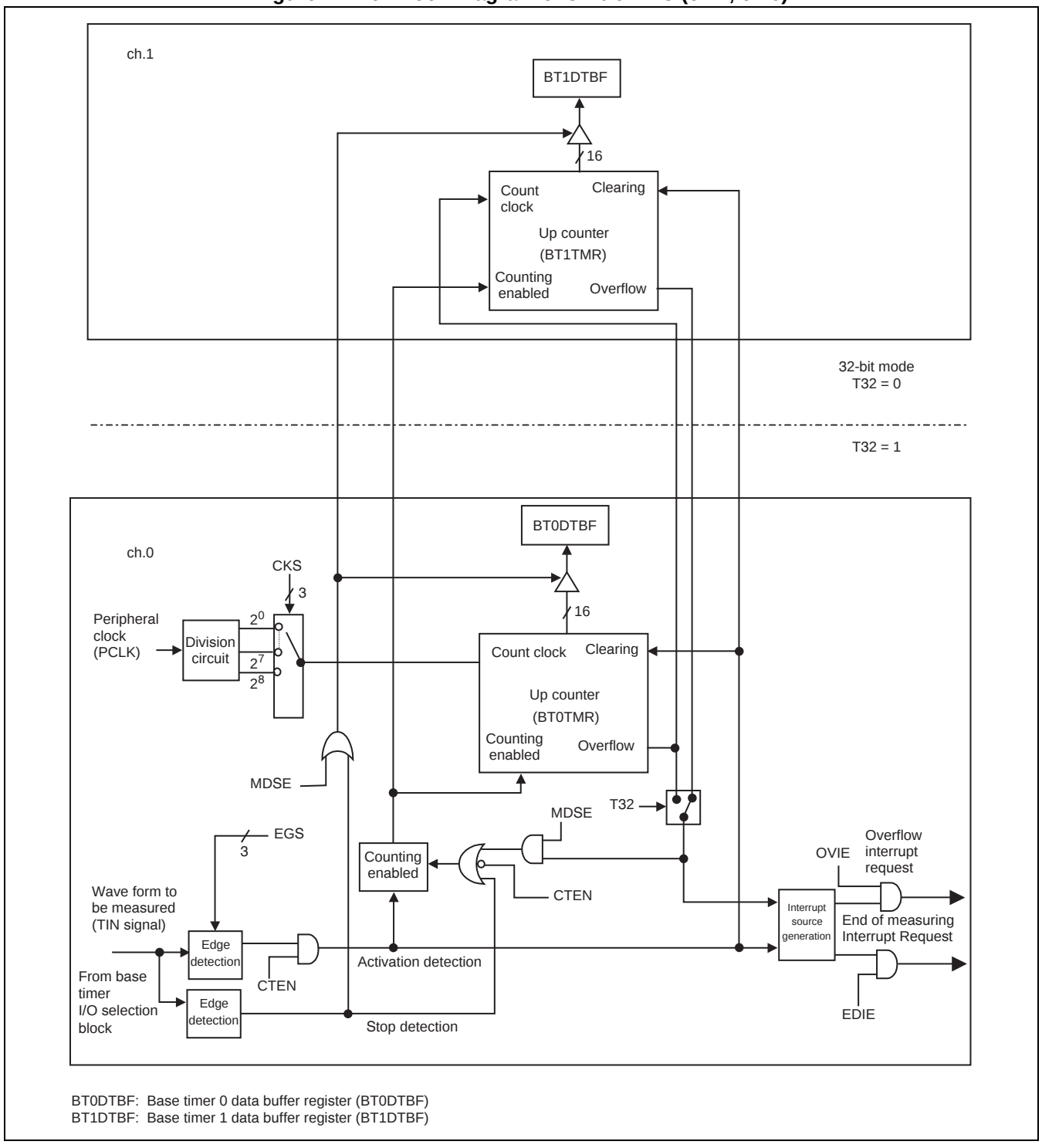
■ Block Diagram of 16/32-bit PWC Timer (ch.1, ch.0)

Figure 24.2-5 Block Diagram of 16-bit PWC (ch.1, ch.0)



BTxDTBF: Base timer x data buffer register (BTxDTBF)

Figure 24.2-6 Block Diagram of 32-bit PWC (ch.1, ch.0)



BT0DTBF: Base timer 0 data buffer register (BT0DTBF)
BT1DTBF: Base timer 1 data buffer register (BT1DTBF)

<Notes>

- The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, between ch.6 and ch.7, between ch.8 and ch.9, between ch.10 and ch.11, between ch.12 and ch.13, and between ch.14 and ch.15. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 23 Base Timer I/O Select Function".

24.3 Base Timer's Registers

This section lists the registers used for the base timer and their bit configurations in each timer function mode.

■ List of Base Timer's Registers

Table 24.3-1 Registers used for 16-bit PWM timer (1 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	23.4.5
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	23.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	23.4.2
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	23.4.3
Common to 12 to 15	BTSELCDEF	Base timer io select register for ch.C/D/E/F	23.4.4
0	BT0TMCR	Base timer 0 timer control register	24.8.1.1
	BT0STC	Base timer 0 status control register	24.8.1.1
	BT0PCSR	Base timer 0 cycle setting register	24.8.1.2
	BT0PDUT	Base timer 0 duty setting register	24.8.1.3
	BT0TMR	Base timer 0 timer register	24.8.1.4
1	BT1TMCR	Base timer 1 timer control register	24.8.1.1
	BT1STC	Base timer 1 status control register	24.8.1.1
	BT1PCSR	Base timer 1 cycle setting register	24.8.1.2
	BT1PDUT	Base timer 1 duty setting register	24.8.1.3
	BT1TMR	Base timer 1 timer register	24.8.1.4
2	BT2TMCR	Base timer 2 timer control register	24.8.1.1
	BT2STC	Base timer 2 status control register	24.8.1.1
	BT2PCSR	Base timer 2 cycle setting register	24.8.1.2
	BT2PDUT	Base timer 2 duty setting register	24.8.1.3
	BT2TMR	Base timer 2 timer register	24.8.1.4

Table 24.3-1 Registers used for 16-bit PWM timer (2 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
3	BT3TMCR	Base timer 3 timer control register	24.8.1.1
	BT3STC	Base timer 3 status control register	24.8.1.1
	BT3PCSR	Base timer 3 cycle setting register	24.8.1.2
	BT3PDUT	Base timer 3 duty setting register	24.8.1.3
	BT3TMR	Base timer 3 timer register	24.8.1.4
4	BT4TMCR	Base timer 4 timer control register	24.8.1.1
	BT4STC	Base timer 4 status control register	24.8.1.1
	BT4PCSR	Base timer 4 cycle setting register	24.8.1.2
	BT4PDUT	Base timer 4 duty setting register	24.8.1.3
	BT4TMR	Base timer 4 timer register	24.8.1.4
5	BT5TMCR	Base timer 5 timer control register	24.8.1.1
	BT5STC	Base timer 5 status control register	24.8.1.1
	BT5PCSR	Base timer 5 cycle setting register	24.8.1.2
	BT5PDUT	Base timer 5 duty setting register	24.8.1.3
	BT5TMR	Base timer 5 timer register	24.8.1.4
6	BT6TMCR	Base timer 6 timer control register	24.8.1.1
	BT6STC	Base timer 6 status control register	24.8.1.1
	BT6PCSR	Base timer 6 cycle setting register	24.8.1.2
	BT6PDUT	Base timer 6 duty setting register	24.8.1.3
	BT6TMR	Base timer 6 timer register	24.8.1.4
7	BT7TMCR	Base timer 7 timer control register	24.8.1.1
	BT7STC	Base timer 7 status control register	24.8.1.1
	BT7PCSR	Base timer 7 cycle setting register	24.8.1.2
	BT7PDUT	Base timer 7 duty setting register	24.8.1.3
	BT7TMR	Base timer 7 timer register	24.8.1.4
8	BT8TMCR	Base timer 8 timer control register	24.8.1.1
	BT8STC	Base timer 8 status control register	24.8.1.1
	BT8PCSR	Base timer 8 cycle setting register	24.8.1.2
	BT8PDUT	Base timer 8 duty setting register	24.8.1.3
	BT8TMR	Base timer 8 timer register	24.8.1.4

Table 24.3-1 Registers used for 16-bit PWM timer (3 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
9	BT9TMCR	Base timer 9 timer control register	24.8.1.1
	BT9STC	Base timer 9 status control register	24.8.1.1
	BT9PCSR	Base timer 9 cycle setting register	24.8.1.2
	BT9PDUT	Base timer 9 duty setting register	24.8.1.3
	BT9TMR	Base timer 9 timer register	24.8.1.4
10	BTATMCR	Base timer 10 timer control register	24.8.1.1
	BTASTC	Base timer 10 status control register	24.8.1.1
	BTAPCSR	Base timer 10 cycle setting register	24.8.1.2
	BTAPDUT	Base timer 10 duty setting register	24.8.1.3
	BTATMR	Base timer 10 timer register	24.8.1.4
11	BTBTMCR	Base timer 11 timer control register	24.8.1.1
	BTBSTC	Base timer 11 status control register	24.8.1.1
	BTBPCSR	Base timer 11 cycle setting register	24.8.1.2
	BTBPDUT	Base timer 11 duty setting register	24.8.1.3
	BTBTMR	Base timer 11 timer register	24.8.1.4
12	BTCTMCR	Base timer 12 timer control register	24.8.1.1
	BTCSTC	Base timer 12 status control register	24.8.1.1
	BTCPCSR	Base timer 12 cycle setting register	24.8.1.2
	BTCPDUT	Base timer 12 duty setting register	24.8.1.3
	BTCTMR	Base timer 12 timer register	24.8.1.4
13	BTDTMCR	Base timer 13 timer control register	24.8.1.1
	BTDSTC	Base timer 13 status control register	24.8.1.1
	BTDPCSR	Base timer 13 cycle setting register	24.8.1.2
	BTDPDUT	Base timer 13 duty setting register	24.8.1.3
	BTDTMR	Base timer 13 timer register	24.8.1.4
14	BTETMCR	Base timer 14 timer control register	24.8.1.1
	BTESTC	Base timer 14 status control register	24.8.1.1
	BTEPCSR	Base timer 14 cycle setting register	24.8.1.2
	BTEPDUT	Base timer 14 duty setting register	24.8.1.3
	BTETMR	Base timer 14 timer register	24.8.1.4

Table 24.3-1 Registers used for 16-bit PWM timer (4 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
15	BTFTMCR	Base timer 15 timer control register	24.8.1.1
	BTFSTC	Base timer 15 status control register	24.8.1.1
	BTFPCSR	Base timer 15 cycle setting register	24.8.1.2
	BTFPDUT	Base timer 15 duty setting register	24.8.1.3
	BTFTMR	Base timer 15 timer register	24.8.1.4

Table 24.3-2 Registers for the 16-bit PPG timer (1 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	23.4.5
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	23.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	23.4.2
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	23.4.3
Common to 12 to 15	BTSELCDEF	Base timer io select register for ch.C/D/E/F	23.4.4
0	BT0TMCR	Base timer 0 timer control register	24.8.2.1
	BT0STC	Base timer 0 status control register	24.8.2.1
	BT0PRLL	Base timer 0 L width setting register	24.8.2.2
	BT0PRLH	Base timer 0 H width setting register	24.8.2.3
	BT0TMR	Base timer 0 timer register	24.8.2.4
1	BT1TMCR	Base timer 1 timer control register	24.8.2.1
	BT1STC	Base timer 1 status control register	24.8.2.1
	BT1PRLL	Base timer 1 L width setting register	24.8.2.2
	BT1PRLH	Base timer 1 H width setting register	24.8.2.3
	BT1TMR	Base timer 1 timer register	24.8.2.4
2	BT2TMCR	Base timer 2 timer control register	24.8.2.1
	BT2STC	Base timer 2 status control register	24.8.2.1
	BT2PRLL	Base timer 2 L width setting register	24.8.2.2
	BT2PRLH	Base timer 2 H width setting register	24.8.2.3
	BT2TM	Base timer 2 timer register	24.8.2.4

Table 24.3-2 Registers for the 16-bit PPG timer (2 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
3	BT3TMCR	Base timer 3 timer control register	24.8.2.1
	BT3STC	Base timer 3 status control register	24.8.2.1
	BT3PRLL	Base timer 3 L width setting register	24.8.2.2
	BT3PRLH	Base timer 3 H width setting register	24.8.2.3
	BT3TMR	Base timer 3 timer register	24.8.2.4
4	BT4TMCR	Base timer 4 timer control register	24.8.2.1
	BT4STC	Base timer 4 status control register	24.8.2.1
	BT4PRLL	Base timer 4 L width setting register	24.8.2.2
	BT4PRLH	Base timer 4 H width setting register	24.8.2.3
	BT4TMR	Base timer 4 timer register	24.8.2.4
5	BT5TMCR	Base timer 5 timer control register	24.8.2.1
	BT5STC	Base timer 5 status control register	24.8.2.1
	BT5PRLL	Base timer 5 L width setting register	24.8.2.2
	BT5PRLH	Base timer 5 H width setting register	24.8.2.3
	BT5TMR	Base timer 5 timer register	24.8.2.4
6	BT6TMCR	Base timer 6 timer control register	24.8.2.1
	BT6STC	Base timer 6 status control register	24.8.2.1
	BT6PRLL	Base timer 6 L width setting register	24.8.2.2
	BT6PRLH	Base timer 6 H width setting register	24.8.2.3
	BT6TMR	Base timer 6 timer register	24.8.2.4
7	BT7TMCR	Base timer 7 timer control register	24.8.2.1
	BT7STC	Base timer 7 status control register	24.8.2.1
	BT7PRLL	Base timer 7 L width setting register	24.8.2.2
	BT7PRLH	Base timer 7 H width setting register	24.8.2.3
	BT7TMR	Base timer 7 timer register	24.8.2.4
8	BT8TMCR	Base timer 8 timer control register	24.8.2.1
	BT8STC	Base timer 8 status control register	24.8.2.1
	BT8PRLL	Base timer 8 L width setting register	24.8.2.2
	BT8PRLH	Base timer 8 H width setting register	24.8.2.3
	BT8TMR	Base timer 8 timer register	24.8.2.4

Table 24.3-2 Registers for the 16-bit PPG timer (3 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
9	BT9TMCR	Base timer 9 timer control register	24.8.2.1
	BT9STC	Base timer 9 status control register	24.8.2.1
	BT9PRLL	Base timer 9 L width setting register	24.8.2.2
	BT9PRLH	Base timer 9 H width setting register	24.8.2.3
	BT9TMR	Base timer 9 timer register	24.8.2.4
10	BTATMCR	Base timer 10 timer control register	24.8.2.1
	BTASTC	Base timer 10 status control register	24.8.2.1
	BTAPRLL	Base timer 10 L width setting register	24.8.2.2
	BTAPRLH	Base timer 10 H width setting register	24.8.2.3
	BTATMR	Base timer 10 timer register	24.8.2.4
11	BTBTMCR	Base timer 11 timer control register	24.8.2.1
	BTBSTC	Base timer 11 status control register	24.8.2.1
	BTBPRLL	Base timer 11 L width setting register	24.8.2.2
	BTBPRLH	Base timer 11 H width setting register	24.8.2.3
	BTBTMR	Base timer 11 timer register	24.8.2.4
12	BTCTMCR	Base timer 12 timer control register	24.8.2.1
	BTCSTC	Base timer 12 status control register	24.8.2.1
	BTCPRLL	Base timer 12 L width setting register	24.8.2.2
	BTCPRLH	Base timer 12 H width setting register	24.8.2.3
	BTCTMR	Base timer 12 timer register	24.8.2.4
13	BTDTMCR	Base timer 13 timer control register	24.8.2.1
	BTDSTC	Base timer 13 status control register	24.8.2.1
	BTDPRLL	Base timer 13 L width setting register	24.8.2.2
	BTDPRLH	Base timer 13 H width setting register	24.8.2.3
	BTDTMR	Base timer 13 timer register	24.8.2.4
14	BTETMCR	Base timer 14 timer control register	24.8.2.1
	BTESTC	Base timer 14 status control register	24.8.2.1
	BTEPRLL	Base timer 14 L width setting register	24.8.2.2
	BTEPRLH	Base timer 14 H width setting register	24.8.2.3
	BTETMR	Base timer 14 timer register	24.8.2.4

Table 24.3-2 Registers for the 16-bit PPG timer (4 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
15	BTFTMCR	Base timer 15 timer control register	24.8.2.1
	BTFSTC	Base timer 15 status control register	24.8.2.1
	BTFPRLL	Base timer 15 L width setting register	24.8.2.2
	BTFPRLH	Base timer 15 H width setting register	24.8.2.3
	BTFTMR	Base timer 15 timer register	24.8.2.4

Table 24.3-3 Registers for the 16/32-bit reload timer (1 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	23.4.5
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	23.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	23.4.2
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	23.4.3
Common to 12 to 15	BTSELCDEF	Base timer io select register for ch.C/D/E/F	23.4.4
0	BT0TMCR	Base timer 0 timer control register	24.8.3.1
	BT0STC	Base timer 0 status control register	24.8.3.1
	BT0PCSR	Base timer 0 cycle setting register	24.8.3.2
	BT0TMR	Base timer 0 timer register	24.8.3.3
1	BT1TMCR	Base timer 1 timer control register	24.8.3.1
	BT1STC	Base timer 1 status control register	24.8.3.1
	BT1PCSR	Base timer 1 cycle setting register	24.8.3.2
	BT1TMR	Base timer 1 timer register	24.8.3.3
2	BT2TMCR	Base timer 2 timer control register	24.8.3.1
	BT2STC	Base timer 2 status control register	24.8.3.1
	BT2PCSR	Base timer 2 cycle setting register	24.8.3.2
	BT2TMR	Base timer 2 timer register	24.8.3.3
3	BT3TMCR	Base timer 3 timer control register	24.8.3.1
	BT3STC	Base timer 3 status control register	24.8.3.1
	BT3PCSR	Base timer 3 cycle setting register	24.8.3.2
	BT3TMR	Base timer 3 timer register	24.8.3.3

Table 24.3-3 Registers for the 16/32-bit reload timer (2 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
4	BT4TMCR	Base timer 4 timer control register	24.8.3.1
	BT4STC	Base timer 4 status control register	24.8.3.1
	BT4PCSR	Base timer 4 cycle setting register	24.8.3.2
	BT4TMR	Base timer 4 timer register	24.8.3.3
5	BT5TMCR	Base timer 5 timer control register	24.8.3.1
	BT5STC	Base timer 5 status control register	24.8.3.1
	BT5PCSR	Base timer 5 cycle setting register	24.8.3.2
	BT5TMR	Base timer 5 timer register	24.8.3.3
6	BT6TMCR	Base timer 6 timer control register	24.8.3.1
	BT6STC	Base timer 6 status control register	24.8.3.1
	BT6PCSR	Base timer 6 cycle setting register	24.8.3.2
	BT6TMR	Base timer 6 timer register	24.8.3.3
7	BT7TMCR	Base timer 7 timer control register	24.8.3.1
	BT7STC	Base timer 7 status control register	24.8.3.1
	BT7PCSR	Base timer 7 cycle setting register	24.8.3.2
	BT7TMR	Base timer 7 timer register	24.8.3.3
8	BT8TMCR	Base timer 8 timer control register	24.8.3.1
	BT8STC	Base timer 8 status control register	24.8.3.1
	BT8PCSR	Base timer 8 cycle setting register	24.8.3.2
	BT8TMR	Base timer 8 timer register	24.8.3.3
9	BT9TMCR	Base timer 9 timer control register	24.8.3.1
	BT9STC	Base timer 9 status control register	24.8.3.1
	BT9PCSR	Base timer 9 cycle setting register	24.8.3.2
	BT9TMR	Base timer 9 timer register	24.8.3.3
10	BTATMCR	Base timer 10 timer control register	24.8.3.1
	BTASTC	Base timer 10 status control register	24.8.3.1
	BTAPCSR	Base timer 10 cycle setting register	24.8.3.2
	BTATMR	Base timer 10 timer register	24.8.3.3

Table 24.3-3 Registers for the 16/32-bit reload timer (3 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
11	BTBTMCR	Base timer 11 timer control register	24.8.3.1
	BTBSTC	Base timer 11 status control register	24.8.3.1
	BTBPCSR	Base timer 11 cycle setting register	24.8.3.2
	BTBTMR	Base timer 11 timer register	24.8.3.3
12	BTCTMCR	Base timer 12 timer control register	24.8.3.1
	BTCSTC	Base timer 12 status control register	24.8.3.1
	BTCPCSR	Base timer 12 cycle setting register	24.8.3.2
	BTCTMR	Base timer 12 timer register	24.8.3.3
13	BTDTMCR	Base timer 13 timer control register	24.8.3.1
	BTDSTC	Base timer 13 status control register	24.8.3.1
	BTDPCSR	Base timer 13 cycle setting register	24.8.3.2
	BTDTMR	Base timer 13 timer register	24.8.3.3
14	BTETMCR	Base timer 14 timer control register	24.8.3.1
	BTESTC	Base timer 14 status control register	24.8.3.1
	BTEPCSR	Base timer 14 cycle setting register	24.8.3.2
	BTETMR	Base timer 14 timer register	24.8.3.3
15	BTFTMCR	Base timer 15 timer control register	24.8.3.1
	BTFSTC	Base timer 15 status control register	24.8.3.1
	BTFPCSR	Base timer 15 cycle setting register	24.8.3.2
	BTFTMR	Base timer 15 timer register	24.8.3.3

Table 24.3-4 List of registers used for 16/32-bit PWC timer (1 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
Common	BTSSSR	Base timer same time soft start register	23.4.5
Common to 0 to 3	BTSEL0123	Base timer io select register for ch.0/1/2/3	23.4.1
Common to 4 to 7	BTSEL4567	Base timer io select register for ch.4/5/6/7	23.4.2
Common to 8 to 11	BTSEL89AB	Base timer io select register for ch.8/9/A/B	23.4.3
Common to 12 to 15	BTSELCDEF	Base timer io select register for ch.C/D/E/F	23.4.4

Table 24.3-4 List of registers used for 16/32-bit PWC timer (2 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
0	BT0TMCR	Base timer 0 timer control register	24.8.4.1
	BT0STC	Base timer 0 status control register	24.8.4.1
	BT0DTBF	Base timer 0 data buffer register	24.8.4.2
1	BT1TMCR	Base timer 1 timer control register	24.8.4.1
	BT1STC	Base timer 1 status control register	24.8.4.1
	BT1DTBF	Base timer 1 data buffer register	24.8.4.2
2	BT2TMCR	Base timer 2 timer control register	24.8.4.1
	BT2STC	Base timer 2 status control register	24.8.4.1
	BT2DTBF	Base timer 2 data buffer register	24.8.4.2
3	BT3TMCR	Base timer 3 timer control register	24.8.4.1
	BT3STC	Base timer 3 status control register	24.8.4.1
	BT3DTBF	Base timer 3 data buffer register	24.8.4.2
4	BT4TMCR	Base timer 4 timer control register	24.8.4.1
	BT4STC	Base timer 4 status control register	24.8.4.1
	BT4DTBF	Base timer 4 data buffer register	24.8.4.2
5	BT5TMCR	Base timer 5 timer control register	24.8.4.1
	BT5STC	Base timer 5 status control register	24.8.4.1
	BT5DTBF	Base timer 5 data buffer register	24.8.4.2
6	BT6TMCR	Base timer 6 timer control register	24.8.4.1
	BT6STC	Base timer 6 status control register	24.8.4.1
	BT6DTBF	Base timer 6 data buffer register	24.8.4.2
7	BT7TMCR	Base timer 7 timer control register	24.8.4.1
	BT7STC	Base timer 7 status control register	24.8.4.1
	BT7DTBF	Base timer 7 data buffer register	24.8.4.2
8	BT8TMCR	Base timer 8 timer control register	24.8.4.1
	BT8STC	Base timer 8 status control register	24.8.4.1
	BT8DTBF	Base timer 8 data buffer register	24.8.4.2
9	BT9TMCR	Base timer 9 timer control register	24.8.4.1
	BT9STC	Base timer 9 status control register	24.8.4.1
	BT9DTBF	Base timer 9 data buffer register	24.8.4.2

Table 24.3-4 List of registers used for 16/32-bit PWC timer (3 / 3)

Channel	Abbreviated Register Name	Register Name	Reference
10	BTATMCR	Base timer 10 timer control register	24.8.4.1
	BTASTC	Base timer 10 status control register	24.8.4.1
	BTADTBF	Base timer 10 data buffer register	24.8.4.2
11	BTBTMCR	Base timer 11 timer control register	24.8.4.1
	BTBSTC	Base timer 11 status control register	24.8.4.1
	BTBDTBF	Base timer 11 data buffer register	24.8.4.2
12	BTCTMCR	Base timer 12 timer control register	24.8.4.1
	BTCSTC	Base timer 12 status control register	24.8.4.1
	BTCDTBF	Base timer 12 data buffer register	24.8.4.2
13	BTDTMCR	Base timer 13 timer control register	24.8.4.1
	BTDSTC	Base timer 13 status control register	24.8.4.1
	BTDDTBF	Base timer 13 data buffer register	24.8.4.2
14	BTETMCR	Base timer 14 timer control register	24.8.4.1
	BTESTC	Base timer 14 status control register	24.8.4.1
	BTEDTBF	Base timer 14 data buffer register	24.8.4.2
15	BTFTMCR	Base timer 15 timer control register	24.8.4.1
	BTFSTC	Base timer 15 status control register	24.8.4.1
	BTFDTBF	Base timer 15 data buffer register	24.8.4.2

24.4 Operations of the Base Timer

This section introduces how the base timer operates in each timer function mode.

■ Operations of the Base Timer

● Reset mode

Placing the base timer in this mode resets its macro (with each register reset to the initial value). Place the base timer in this mode once before changing its function mode or T32 bit setting. After a reset, however, the base timer can set its function mode and the T32 bit without entering the reset mode in advance. If you set this mode for even-numbered channels in 32-bit mode, odd-numbered channels are reset as well at the same time. Thus you do not have to set the reset mode for odd-numbered channels.

● 16-bit PWM timer

The 16-bit PWM timer starts decrementing its counter by the value set as a period when triggered to start. The PWM timer then sets the output to the "L" level first and, if the 16-bit down counter value matches the value set in the duty setting register, inverts the output to the "H" level. Then it inverts the output back to the "L" level when the counter causes an underflow subsequently. This generates a waveform with an arbitrary period and duty cycle.

● 16-bit PPG timer

The 16-bit PPG timer starts decrementing its counter by the value set in the "L"-width setting reload register when triggered to start. The PPG timer then sets the output to the "L" level first and inverts the output back to the "H" level when the counter causes an underflow. The PPG timer continuously decrements the counter by the value set in the "H"-width setting reload register and inverts the output level to "L" when the counter causes an underflow. This generates a waveform with arbitrary "L" and "H" widths.

● 16-bit reload timer

The 16-bit reload timer starts decrementing its 16-bit down counter by the value set as a period when triggered to start. When the down counter causes an underflow, the interrupt flag is set. Depending on the MDSE bit setting, the output level either toggles, or is inverted, between "H" and "L" each time the counter causes an underflow or becomes "H" when the counter starts counting and "L" when it causes an underflow.

● 32-bit reload timer

The 32-bit reload timer is the same in basic operation as the 16-bit reload timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit timers, respectively, interrupt control and output wave control follow their respective settings for the even-numbered channel. To set the period, write the value to the upper register (odd-numbered channel) first and then to the lower register (even-numbered channel).

To obtain the timer value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

<Notes>

- The reload timers can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, between ch.6 and ch.7, between ch.8 and ch.9, between ch.10 and ch.11, between ch.12 and ch.13, and between ch.14 and ch.15. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 23 Base Timer I/O Select Function".

● 16-bit PWC timer

The 16-bit PWC timer starts the 16-bit up counter upon input of a pre-set measurement start edge and stops the counter upon detection of a measurement stop edge. The count value between the two edges is written to the data buffer register as a pulse width.

● 32-bit PWC timer

The 32-bit PWC timer is the same in basic operation as the 16-bit PWC timer, except that it works as a 32-bit version using a pair of even-numbered and odd-numbered channels. Although the even-numbered and odd-numbered channels then operate as the lower 16-bit and upper 16-bit counters, respectively, interrupt control follows the setting for the even-numbered channel. To obtain the measured value or count value, read the lower register (even-numbered channel) first and then the upper register (odd-numbered channel).

<Notes>

- The PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, between ch.6 and ch.7, between ch.8 and ch.9, between ch.10 and ch.11, between ch.12 and ch.13, and between ch.14 and ch.15. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 23 Base Timer I/O Select Function".

24.5 32-bit Mode Operations

The reload timer and PWC timer can operate in 32-bit mode using a pair of channels. This section describes the basic functions and operations of 32-bit mode.

■ Functions of 32-bit Mode

The 32-bit mode combines two channels of base timer into a 32-bit data reload timer or PWC timer. Either 32-bit timer allows the timer/counter value to be read even during operation as it takes the upper 16-bit timer/counter value of the odd-numbered channel also when reading the lower 16-bit timer/counter value of the even-numbered channel.

■ Setting the 32-bit Mode

First, set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register for the even-numbered channel to "000_B" to reset in reset mode. Then, select the reload timer or PWC timer and set its operations in the same way as in 16-bit mode. At this time, write "1" to the T32 bit in the BTxTMCR register to enter the 32-bit operation mode. The T32 bit for the odd-numbered channel must be left containing "0". Neither the reset mode setting is required for the odd-numbered channel. To use the base timer as the reload timer, set the period setting register for the odd-numbered channel to the upper 16-bit reload value among 32 bits and set the period setting register for the even-numbered channel to the lower 16-bit reload value.

As the transition to 32-bit operation mode takes place the moment is written to the T32 bit, the setting must be changed with counting halted on both of the channels.

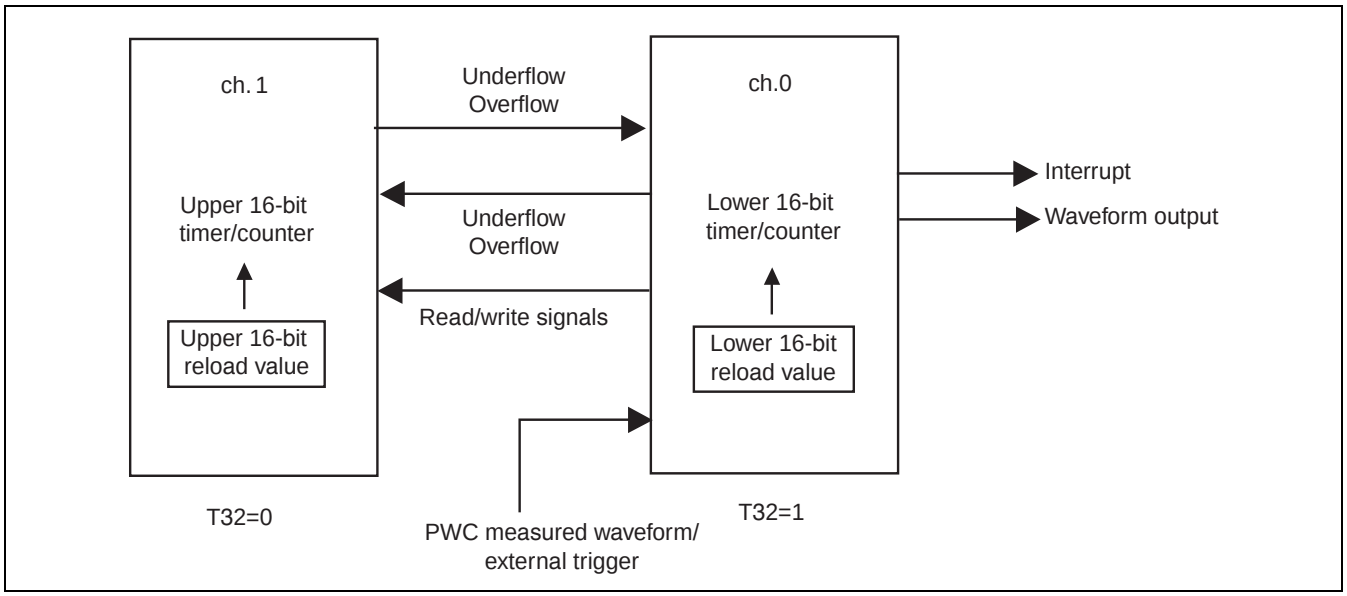
To switch from 32-bit mode to 16-bit mode, set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register for the even-numbered channel to "000_B" to reset the states of both of the even-numbered and odd-numbered channels in reset mode. Then set each channel for operation in 16-bit mode.

■ Operations in 32-bit Mode

When the reload timer or PWC timer is started in 32-bit mode under control of the even-numbered channel, the timer/counter of the even-numbered channel operates as the lower 16-bit timer/counter and the timer/counter of the odd-numbered channel operates as the upper 16-bit one.

In 32-bit mode, the base timer follows the settings for the even-numbered channel while ignoring those for the odd-numbered channel (except the period setting register when serving as the reload timer). Even for the timer start, waveform output, and interrupt signal settings, the even-numbered channel overrides the odd-numbered channel (odd-numbered channel is always masked at "L").

The following example shows a PWC configuration using ch.0 and ch.1.



<Notes>

- The reload timer or PWC timer can operate in 32 bits only between ch.0 and ch.1, between ch.2 and ch.3, between ch.4 and ch.5, between ch.6 and ch.7, between ch.8 and ch.9, between ch.10 and ch.11, between ch.12 and ch.13, and between ch.14 and ch.15. No 32-bit operation is applicable to any other combination of channels.
- This function supports simultaneous activation. For details, see "CHAPTER 23 Base Timer I/O Select Function".

24.6 Notes of Using the Base Timer

This section summarizes the notes on using the base timer.

■ Common Notes on Using Each Type of Timer

● Notes on setting through programming

- The following bits in the BTxTMCR register must not be updated during operation. Be sure to update them before starting the base timer or after stopping it.

[bit14, bit13, bit12]	CKS2, CKS1, CKS0	: Clock select bits
[bit10, bit9, bit8]	EGS2, EGS1, EGS0	: Measurement edge select bits
[bit7]	T32	: 32-bit timer select bit (Used with the reload timer or PWC timer selected)
[bit6, bit5,bit4]	FMD2, FMD1, FMD0	: Timer function mode select bits
[bit2]	MDSE	: Measurement mode (one-shot/continuous) select bit

- If you set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to "000_B" to enter the reset mode, all the registers of the base timer are initialized and thus they must be set all over again.
- If you set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to "000_B" to enter the reset mode, the other bits in the BTxTMCR register are initialized with their settings ignored.

■ Notes on Using the 16-bit PWM/PPG/Reload Timer

● Notes on setting through programming

- When the interrupt request flag is attempted to be set and cleared at the same timing, the flag set action overrides the flag clear action.
- When the down counter is attempted to load and count at the same timing, the load action overrides the count action.
- Set the FMD2, FMD1, and FMD0 bits in the BTxTMCR register to select the timer function mode before setting the period, duty cycle, "H" width, and "L" width.
- If a restart is detected when counting is completed in one-shot mode, the counter is restarted with the count value reloaded.

■ Notes on Using the PWC Timer

● Notes on setting through programming

- Writing "1" to the counting enable bit (CTEN) clears the counter, nullifying the data existing in the counter before counting is enabled.
- If you set the PWC mode (FMD = 100_B) after a system reset or in reset mode and enables measurement (CTEN = 1) at the same time, the timer may operate according to the immediately preceding measurement signal.
- If a measurement start edge is detected the moment a restart is set in continuous measurement mode, the timer immediately starts counting from "0001_H".
- An attempt to restart the timer after starting counting can result as follows, depending on that timing:
- If the attempt is made at a measurement end edge in one-shot pulse width measurement mode: Although the timer is restarted and waits for an measurement start edge, the measurement end flag (EDIR) is set.
- If the attempt is made at a measurement end edge in continuous pulse width measurement mode: Although the timer is restarted and waits for a measurement start edge, the measurement end flag (EDIR) is set and the current measurement result is transferred to the BTxDTBF register.

When restarting the timer during operation, control interrupts while paying attention to the behaviors of flags.

24.7 Base Timer Interrupts

This section lists the interrupt request flags, interrupt enable bits, and interrupt factors for the base timer in each timer function mode.

■ Interrupt Control Bits and Interrupt Factors by Timer Function Mode

Table 24.7-1 lists the interrupt control bits and interrupt factors for the base timer in each timer function mode.

Table 24.7-1 Interrupt Control Bits and Interrupt Factors in Each Timer Function Mode

	Status control register (BTxSTC)			
	Interrupt request flag bits	**Interrupt request enable bits**	**Interrupt factors**	**IRQ**
PWM timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	DTIR: bit1	DTIE: bit5	Duty match detection	
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PPG timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
Reload timer function	UDIR: bit0	UDIE: bit4	Underflow detection	IRQ0
	TGIR: bit2	TGIE: bit6	Timer start trigger detection	IRQ1
PWC timer function	OVIR: bit0	OVIE: bit4	Overflow detection	IRQ0
	EDIR: bit2	EDIE: bit6	Measurement end detection	IRQ1

24.8 Base Timer Description by Function Mode

This section describes each function of the base timer.

■ Base Timer Function

- PWM function
- PPG function
- Reload timer function
- PWC function

24.8.1 PWM Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWM timer.

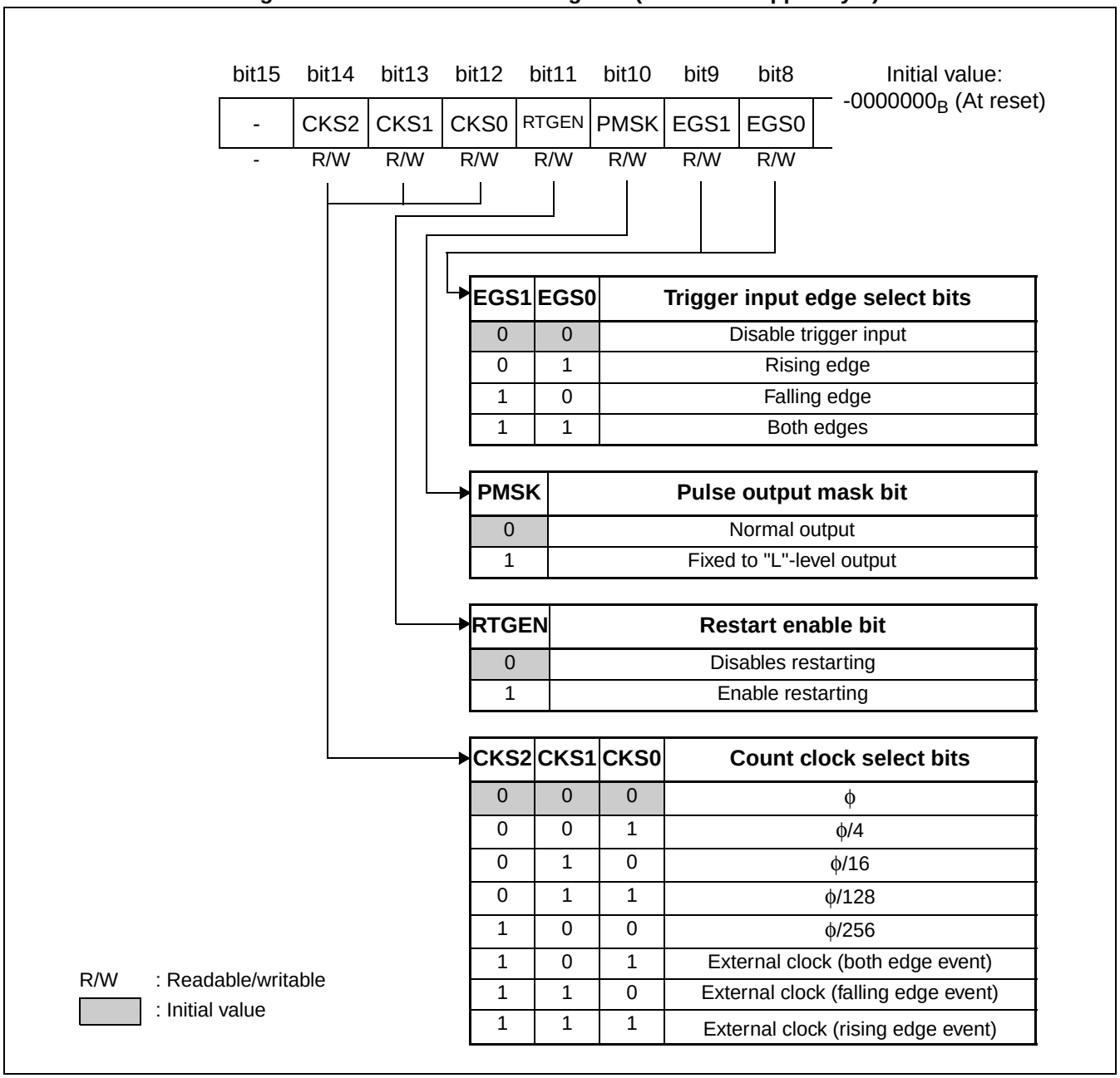
- Timer Control Register (BTxTMCR) for PWM Timer
- PWM Period Setting Register (BTxPCSR)
- PWM Duty Setting Register (BTxPDUT)
- Timer Register (BTxTMR)
- 16-bit PWM Timer Operation
- One-shot Operation
- Interrupt Factors and Timing Chart
- Output Waveforms

24.8.1.1 Timer Control Register (BTxTMCR) for PWM Timer

The timer control register (BTxTMCR) controls the PWM timer. Keep in mind that the register contains bits which cannot be updated with the PWM timer operating.

■ Timer Control Register (BTxTMCR Upper Byte)

Figure 24.8-1 Timer Control Register (BTxTMCR Upper Byte)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
-0000000B (At reset)
- CKS2 CKS1 CKS0 RTGEN PMSK EGS1 EGS0
- R/W R/W R/W R/W R/W R/W R/W
R/W : Readable/writable
: Initial value

EGS1	EGS0	Trigger input edge select bits
0	0	Disable trigger input
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

PMSK	Pulse output mask bit
0	Normal output
1	Fixed to "L"-level output

RTGEN	Restart enable bit
0	Disables restarting
1	Enable restarting

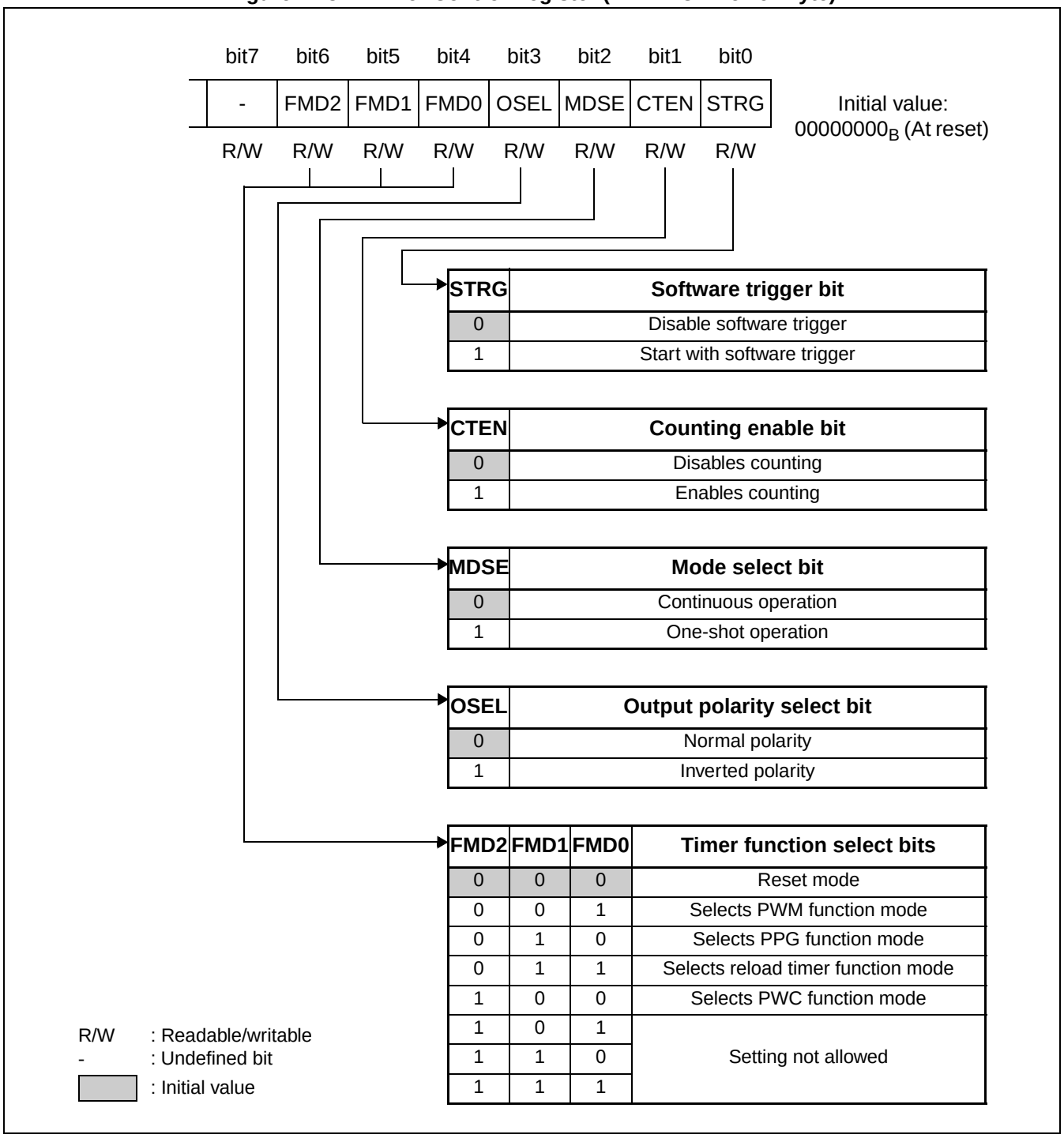
CKS2	CKS1	CKS0	Count clock select bits
0	0	0	ϕ
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	External clock (both edge event)
1	1	0	External clock (falling edge event)
1	1	1	External clock (rising edge event)

Table 24.8-1 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	• The read value of this bit is undefined. • Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	• Select the count clock for the 16-bit down counter. • The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	Enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	• Controls the PWM output waveform level. • When this bit is "0", the PWM waveform is output as it is. • When the bit is "1", the PWM output is masked to the "L" level irrespective of the period and duty cycle. Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PWM output to the "H" level.
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	• Select the effective edge of the input waveform as an external trigger to set the trigger condition. • When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. • EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTxTMCR Lower Byte)

Figure 24.8-2 Timer Control Register (BTxTMCR Lower Byte)


bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
- FMD2 FMD1 FMD0 OSEL MDSE CTEN STRG
Initial value: 00000000B (At reset)
R/W R/W R/W R/W R/W R/W R/W R/W
R/W : Readable/writable
- : Undefined bit
: Initial value

STRG	Software trigger bit
0	Disable software trigger
1	Start with software trigger

CTEN	Counting enable bit
0	Disables counting
1	Enables counting

MDSE	Mode select bit
0	Continuous operation
1	One-shot operation

OSEL	Output polarity select bit
0	Normal polarity
1	Inverted polarity

FMD2	FMD1	FMD0	Timer function select bits
0	0	0	Reset mode
0	0	1	Selects PWM function mode
0	1	0	Selects PPG function mode
0	1	1	Selects reload timer function mode
1	0	0	Selects PWC function mode
1	0	1	Setting not allowed
1	1	0	
1	1	1	

Table 24.8-2 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function
bit7	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit6 to bit4	FMD2, FMD1, FMD0: Timer function select bits	• These bits select the timer function mode. • Setting the FMD2, FMD1, and FMD0 bits to "001_B" selects the PWM function mode. • The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit3	OSEL: Output polarity select bit	Selects the polarity of PWM output. Polarity / After reset / Duty match / Underflow Normal / "L" output / / Inverted / "H" output / /
bit2	MDSE: Mode select bit	• Selects continuous pulse output or one-shot pulse output. • The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit1	CTEN: Counting enable bit	• This bit enables the down counter. • Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.
bit0	STRG: Software trigger bit	• Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger. Note: Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger. • The value read from the STRG bit is always "0". Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.

■ Status Control Register (BTxSTC)

Figure 24.8-3 Status Control Register (BTxSTC)

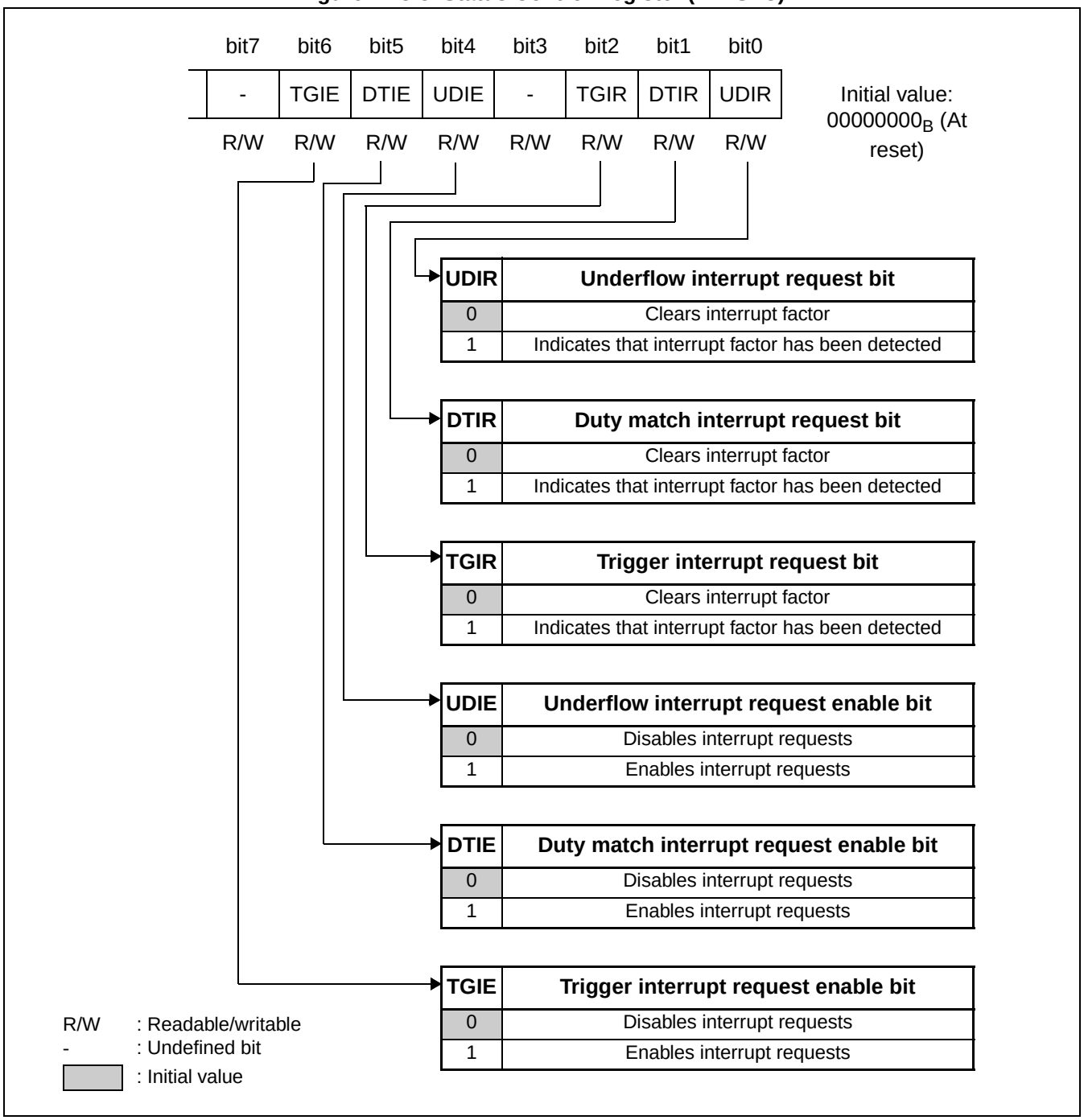

bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
- TGIE DTIE UDIE - TGIR DTIR UDIR
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value: 00000000B (At reset)
UDIR Underflow interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
DTIR Duty match interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
TGIR Trigger interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
UDIE Underflow interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
DTIE Duty match interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
TGIE Trigger interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-3 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	• Controls bit2: TGIR interrupt requests. • Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	DTIE: Duty match interrupt request enable bit	• Controls bit1: DTIR interrupt requests. • Setting the DTIR bit (bit1) with the DTIE bit enabling duty match interrupt requests generates an interrupt request to the CPU.
bit4	UDIE: Underflow interrupt request enable bit	• Controls bit0: UDIR interrupt requests. • Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	• The TGIR bit is set to "1" upon detection of a software trigger or trigger input. • Writing "0" to the TGIR bit clears it. • Writing "1" to the TGIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	DTIR: Duty match interrupt request bit	• The DTIR bit is set to "1" when the count value matches the duty cycle setting. • Writing "0" to the DTIR bit clears it. • Writing "1" to the DTIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit0	UDIR: Underflow interrupt request bit	• The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$. • Writing "0" to the UDIR bit clears it. • Writing "1" to the UDIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

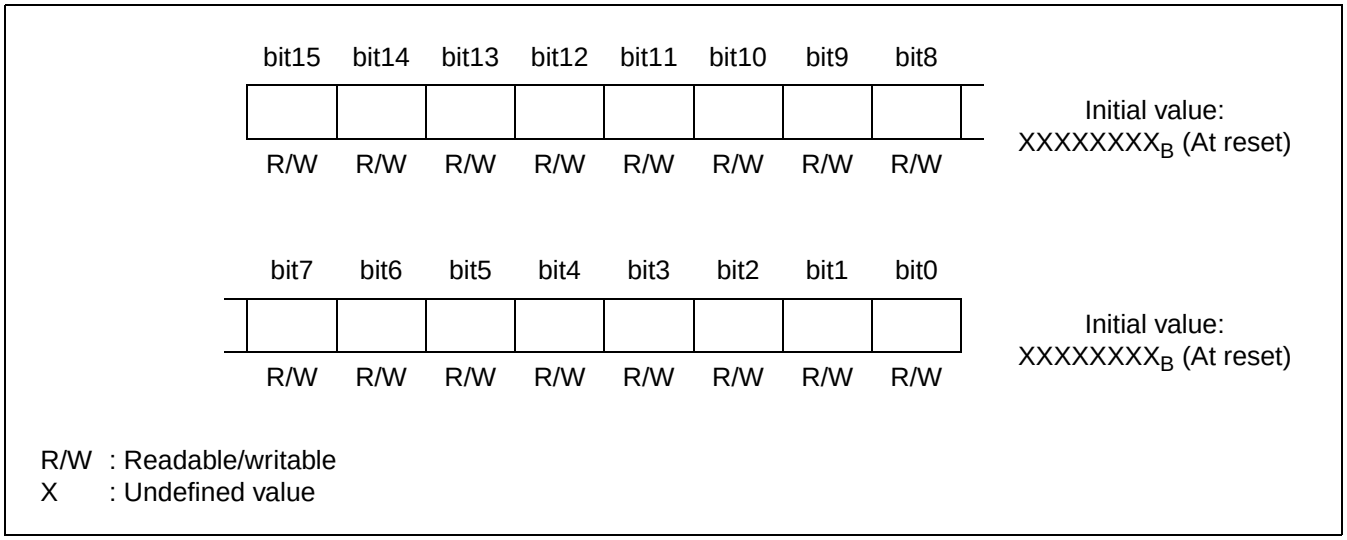
24.8.1.2 PWM Period Setting Register (BTxPCSR)

The PWM period setting register (BTxPCSR) is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

■ Bit Configuration of the PWM Period Setting Register (BTxPCSR)

Figure 24.8-4 shows the bit configuration of the PWM period setting register (BTxPCSR).

Figure 24.8-4 Bit Configuration of the PWM Period Setting Register (BTxPCSR)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
R/W : Readable/writable
X : Undefined value

The BTxPCSR register is a buffered register for setting the PWM period. Transfer to the timer register takes place when the counter is started and when it causes an underflow.

After writing to the period setting register to initially set or update it, be sure to write to the duty setting register.

- Access the BTxPCSR register using 16-bit data.
- Set the PWM period using the BTxPCSR register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

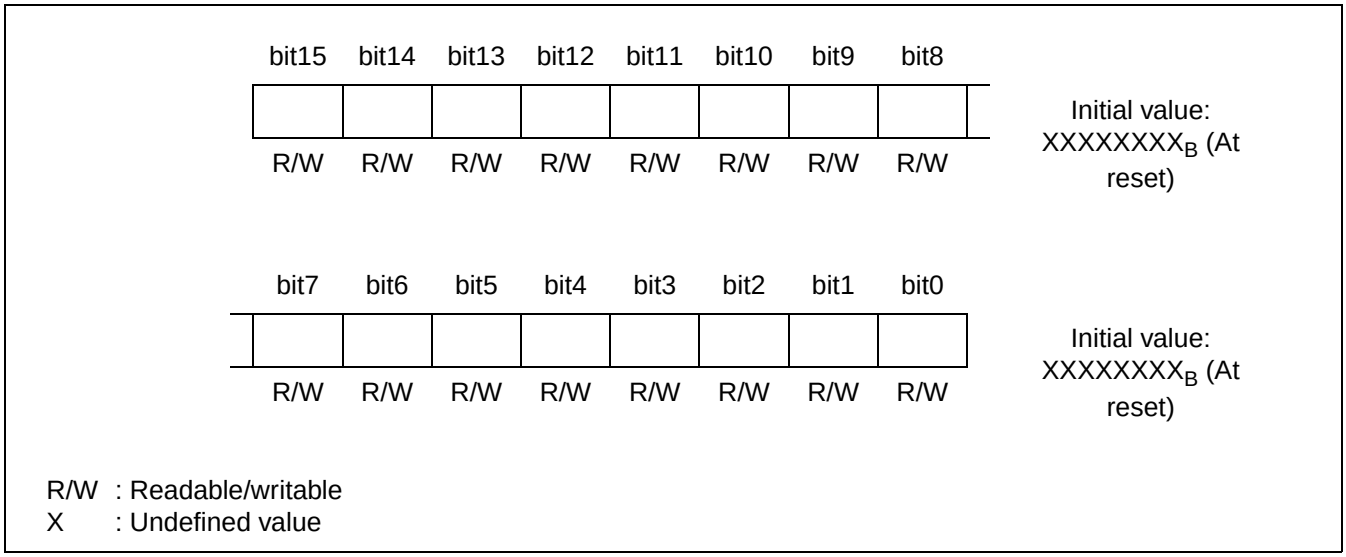
24.8.1.3 PWM Duty Setting Register (BTxPDUT)

The PWM duty setting register (BTxPDUT) is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

■ Bit Configuration of the PWM Duty Setting Register (BTxPDUT)

Figure 24.8-5 shows the bit configuration of the PWM duty setting register (BTxPDUT).

Figure 24.8-5 Bit Configuration of the PWM Duty Setting Register (BTxPDUT)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value: XXXXXXXX$_B$ (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value: XXXXXXXX$_B$ (At reset)
R/W : Readable/writable
X : Undefined value

The BTxPDUT register is a buffered register for setting the PWM duty cycle. Transfer from the buffer takes place when an underflow occurs.

If you set the period setting and duty setting registers to the same value, the output level is all "H" in normal polarity or all "L" in inverted polarity.

Do not set the BTxPDUT register to a value greater than the value of the PSCR register, or PWM output will be undefined.

- Access the BTxPDUT register using 16-bit data.
- Set the PWM duty cycle using the BTxPDUT register after selecting the PWM function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

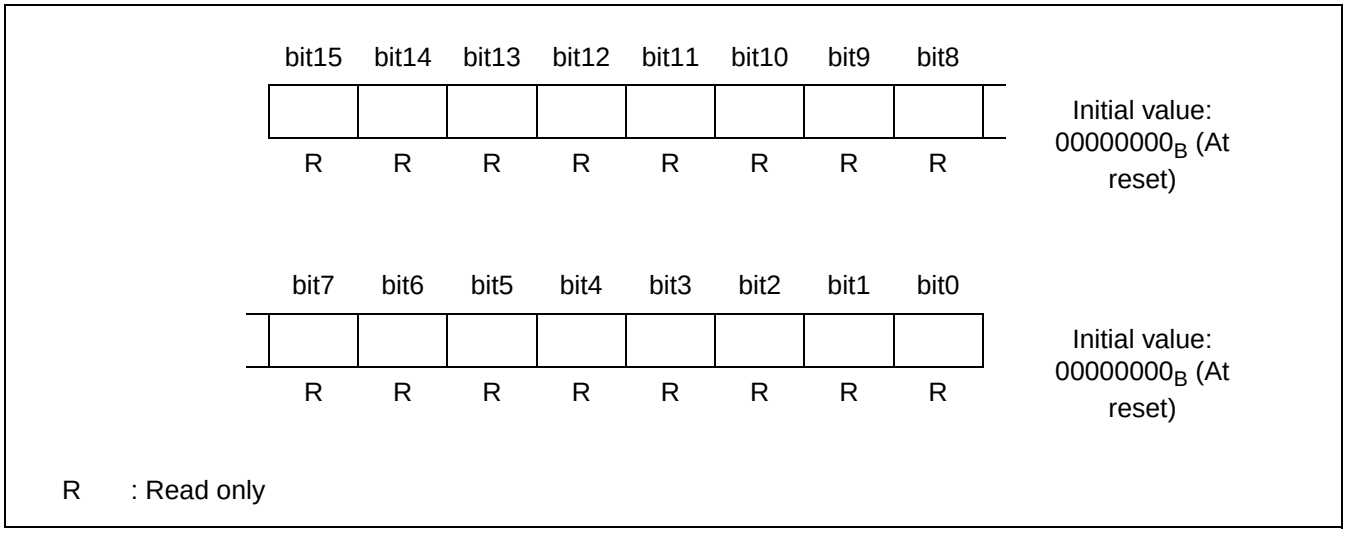
24.8.1.4 Timer Register (BTxTMR)

The timer register (BTxTMR) allows the value of the 16-bit down counter to be read from.

■ Bit Configuration of the Timer Register (BTxTMR)

Figure 24.8-6 shows the bit configuration of the PWM timer register (BTxTMR).

Figure 24.8-6 Bit Configuration of the Timer Register (BTxTMR)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R R R R R R R R
Initial value: 00000000B (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R R R R R R R R
Initial value: 00000000B (At reset)
R : Read only

The BTxTMR register allows the value of the 16-bit down counter to be read from.

<Note>

Access the BTxTMR register using 16-bit data.

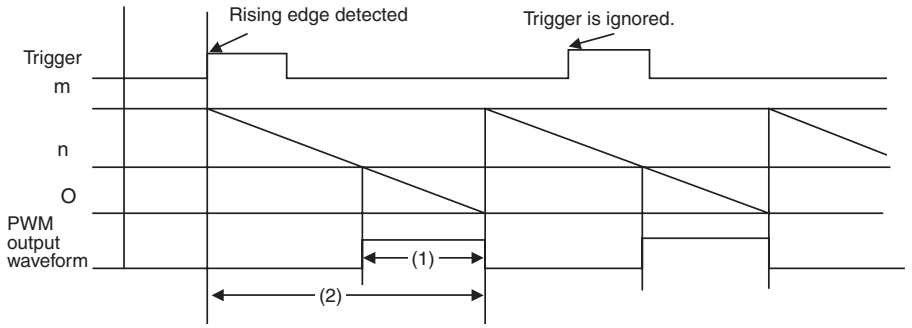
24.8.1.5 16-bit PWM Timer Operation

In PWM timer mode, a waveform having a specified period can be output either in single shots or continuously after detection of a trigger.
The period of output pulses can be controlled by changing the BTxPCSR value.
The duty ratio can be controlled by changing the BTxPDUT value. After writing data to the BTxPCSR register, be sure to write to the BTxPDUT register as well.

■ Continuous Operation

● When restarting is disabled (RTGEN = 0)

Figure 24.8-7 PWM Operation Timing Chart (Restarting Disabled)

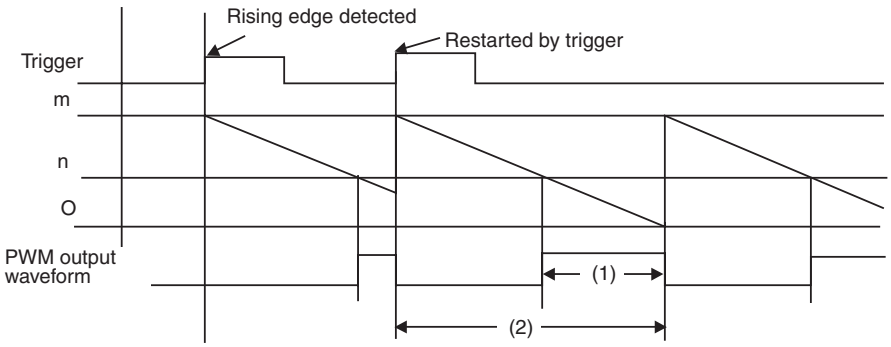


(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

● When restarting is enabled (RTGEN = 1)

Figure 24.8-8 PWM Operation Timing Chart (Restarting Enabled)



(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

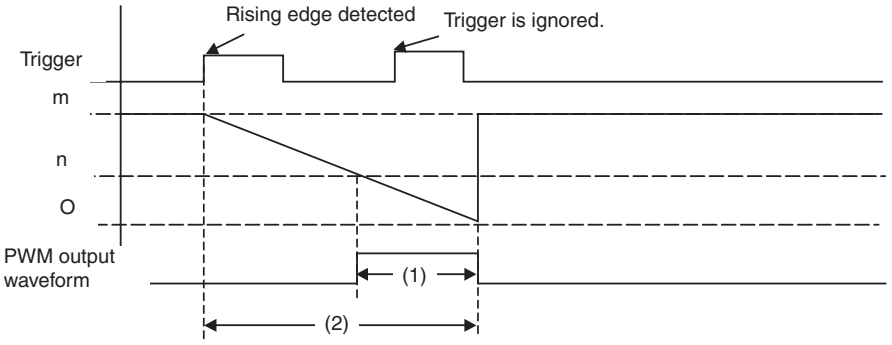
24.8.1.6 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ One-shot Operation

● When restarting is disabled (RTGEN = 0)

Figure 24.8-9 One-shot Operation Timing Chart (Trigger Restarting Disabled)

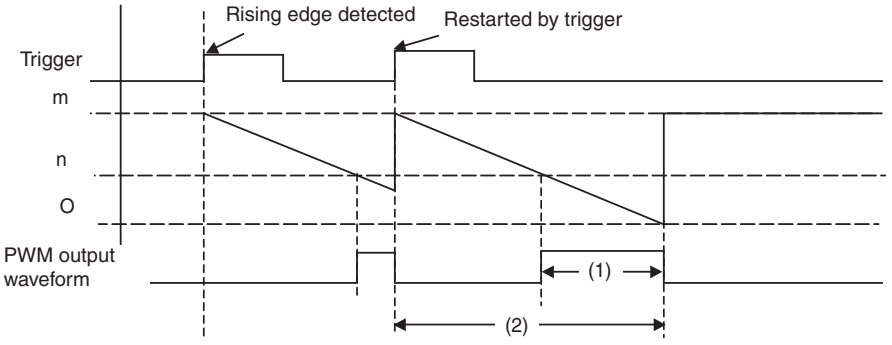


(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

● When restarting is enabled (RTGEN = 1)

Figure 24.8-10 One-shot Operation Timing Chart (Trigger Restarting Enabled)



(1) = T(n+1) ms
(2) = T(m+1) ms

T : Count clock cycle
m : BTxPCSR value
n : BTxPDUT value

24.8.1.7 Interrupt Factors and Timing Chart

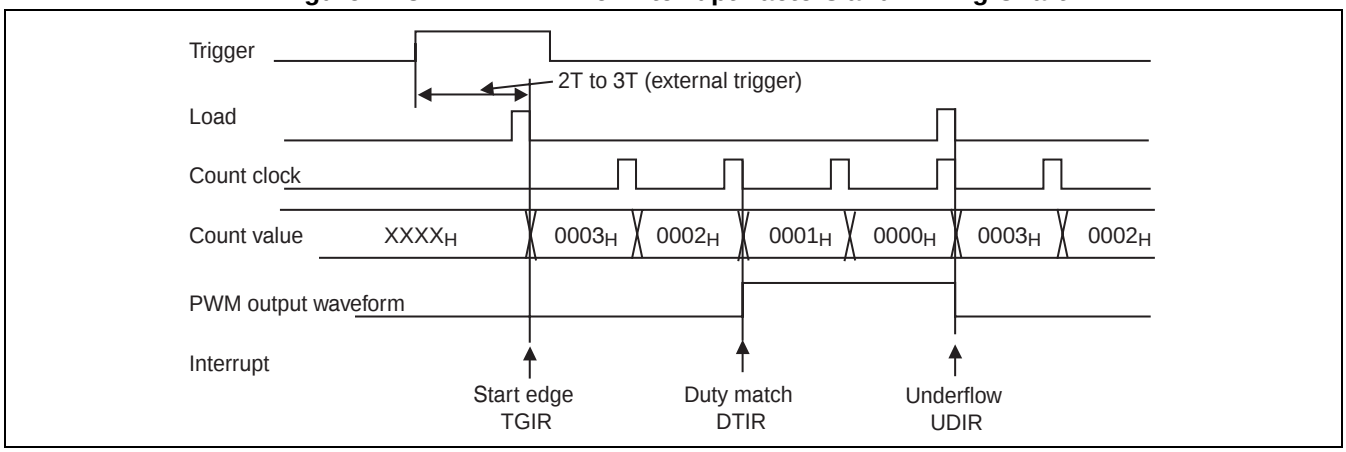
This section provides the interrupt factors and timing chart.

■ Interrupt Factors and Timing Chart (PWM Output: Normal Polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (PCLK) cycle) until the counter value is loaded after the input of the trigger.

Figure 24.8-11 shows the interrupt factors and timing chart, assuming "period setting" = 3 and "duty value" = 1.

Figure 24.8-11 PWM Timer Interrupt Factors and Timing Chart



24.8.1.8 Output Waveforms

This section illustrates PWM output.

■ PWM Output at All "L" or All "H" Level

Figure 24.8-12 and Figure 24.8-13 illustrate how to provide PWM output at all "L" and all "H" levels, respectively.

Figure 24.8-12 Example of PWM Output at All "L" Level

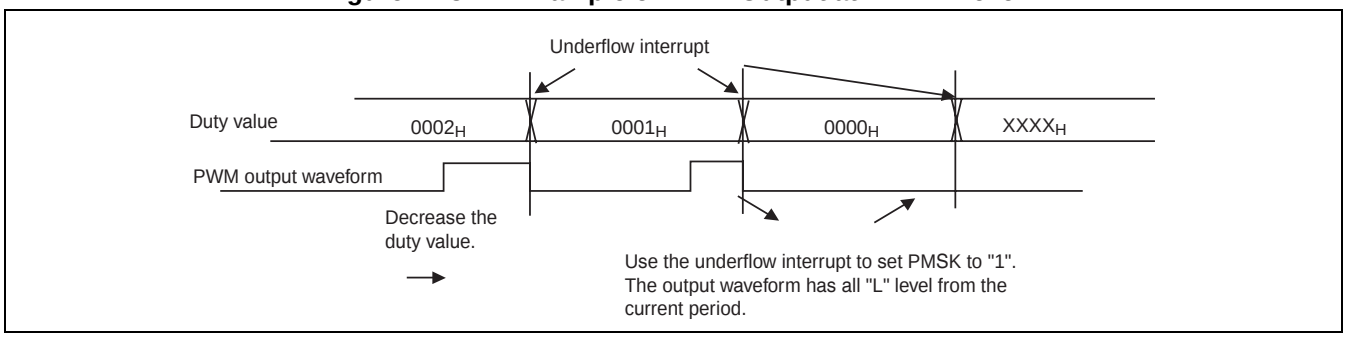
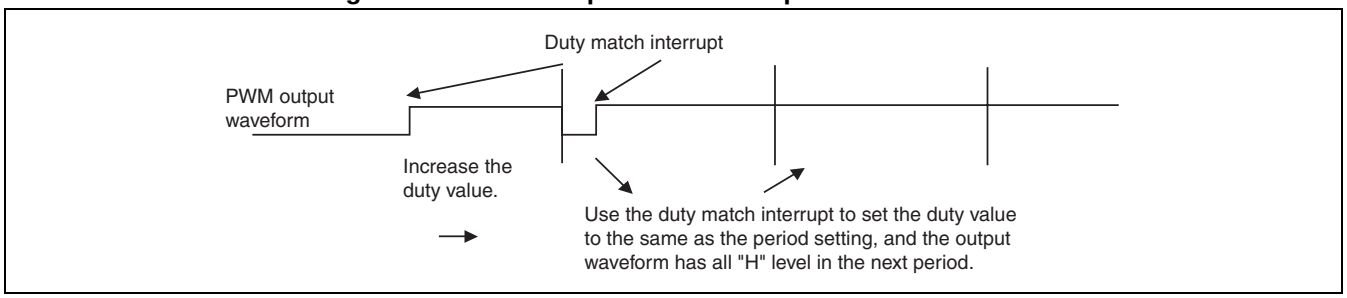


Figure 24.8-13 Example of PWM Output at All "H" Level


Duty match interrupt
PWM output waveform
Increase the duty value.
Use the duty match interrupt to set the duty value to the same as the period setting, and the output waveform has all "H" level in the next period.

24.8.2 PPG Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PPG timer.

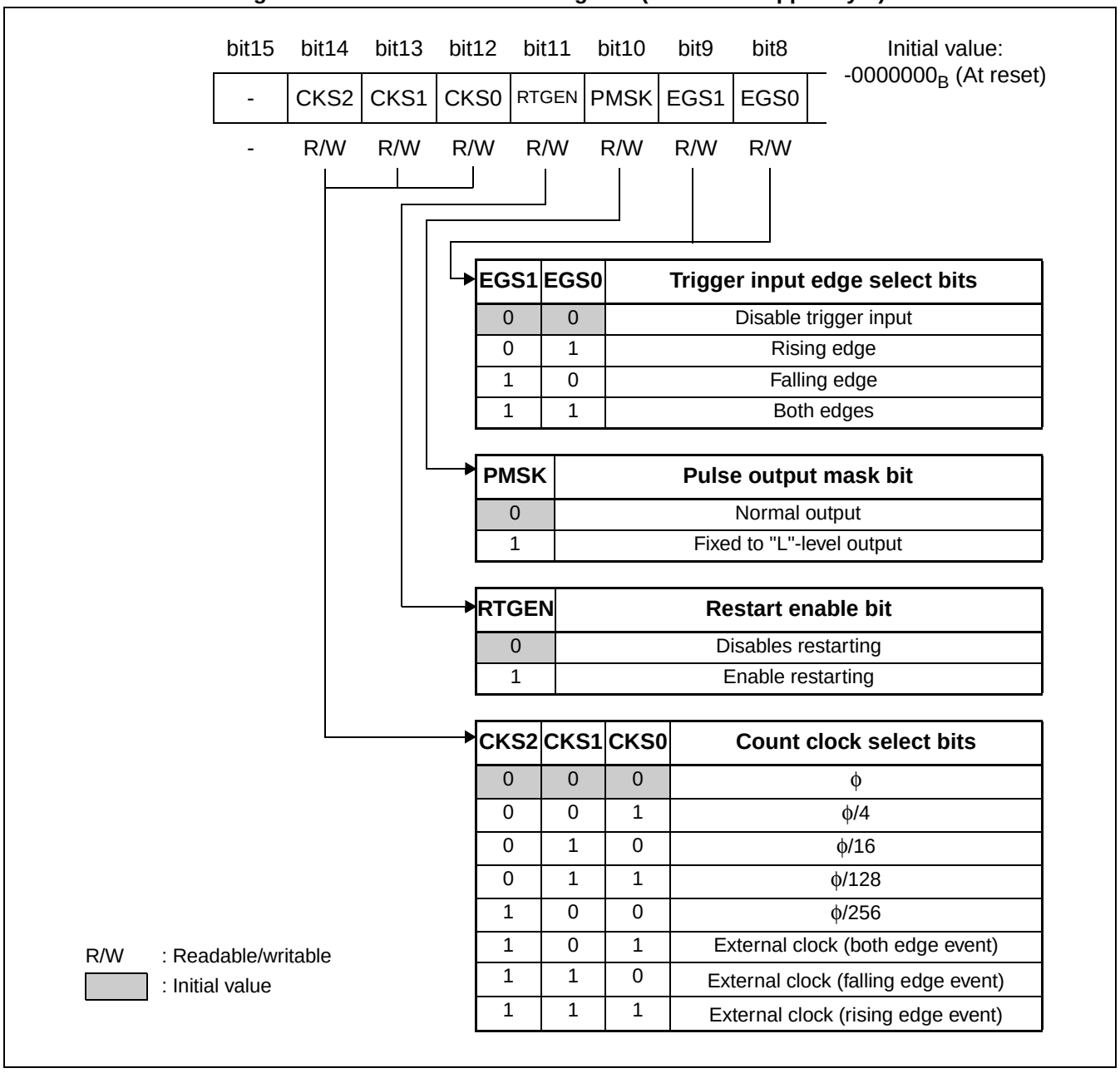
- Timer Control Register (BTxTMCR) for PPG Timer
- "L"-width Setting Reload Register (BTxPRLL)
- "H"-width Setting Reload Register (BTxPRLH)
- Timer Register (BTxTMR)
- 16-bit PPG Timer Operation
- Continuous Operation
- One-shot Operation
- Interrupt Factors and Timing Chart

24.8.2.1 Timer Control Register (BTxTMCR) for PPG Timer

The timer control register (BTxTMCR) controls the PPG timer. Keep in mind that the register contains bits which cannot be updated with the PPG timer operating.

■ Timer Control Register (BTxTMCR Upper Byte)

Figure 24.8-14 Timer Control Register (BTxTMCR Upper Byte)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
-0000000B (At reset)
- CKS2 CKS1 CKS0 RTGEN PMSK EGS1 EGS0
- R/W R/W R/W R/W R/W R/W R/W
R/W : Readable/writable
: Initial value

EGS1	EGS0	Trigger input edge select bits
0	0	Disable trigger input
0	1	Rising edge
1	0	Falling edge
1	1	Both edges

PMSK	Pulse output mask bit
0	Normal output
1	Fixed to "L"-level output

RTGEN	Restart enable bit
0	Disables restarting
1	Enable restarting

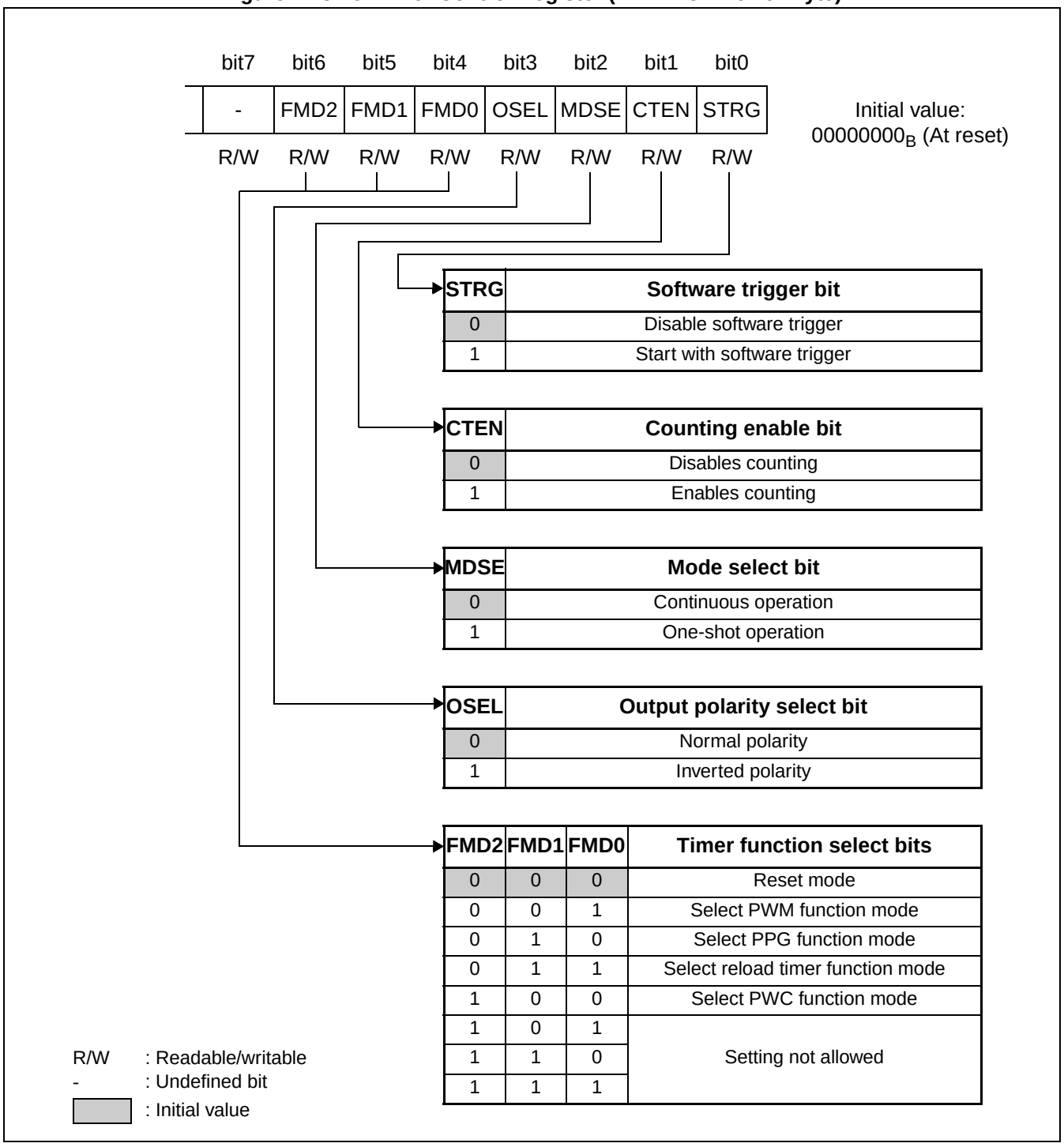
CKS2	CKS1	CKS0	Count clock select bits
0	0	0	ϕ
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	External clock (both edge event)
1	1	0	External clock (falling edge event)
1	1	1	External clock (rising edge event)

Table 24.8-4 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	• The read value of this bit is undefined. • Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	• Select the count clock for the 16-bit down counter. • The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	RTGEN: Restart enable bit	This bit enables restarting with a software trigger or trigger input.
bit10	PMSK: Pulse output mask bit	• Controls the PPG output waveform level. • When this bit is "0", the PPG waveform is output as it is. • When the bit is "1", the PPG output is masked to the "L" level irrespective of the "H" and "L" width settings. Note: Setting the PMSK bit to "1" with the OSEL bit (bit3) set for inverted output masks the PPG output to the "H" level.
bit9, bit8	EGS1, EGS0: Trigger input edge select bits	• Select the effective edge of the input waveform as an external trigger to set the trigger condition. • When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. • EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTxTMCR Lower Byte)

Figure 24.8-15 Timer Control Register (BTxTMCR Lower Byte)


bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
- FMD2 FMD1 FMD0 OSEL MDSE CTEN STRG
Initial value:
00000000B (At reset)
R/W R/W R/W R/W R/W R/W R/W R/W

STRG	Software trigger bit
0	Disable software trigger
1	Start with software trigger

CTEN	Counting enable bit
0	Disables counting
1	Enables counting

MDSE	Mode select bit
0	Continuous operation
1	One-shot operation

OSEL	Output polarity select bit
0	Normal polarity
1	Inverted polarity

FMD2	FMD1	FMD0	Timer function select bits
0	0	0	Reset mode
0	0	1	Select PWM function mode
0	1	0	Select PPG function mode
0	1	1	Select reload timer function mode
1	0	0	Select PWC function mode
1	0	1	Setting not allowed
1	1	0	
1	1	1	

R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-5 Timer Control Register (BTxTMCR Lower Byte)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>Undefined bit</td><td>• The value read is "0"
• When writing to this bit, write "0".</td></tr>
<tr><td>bit6
to
bit4</td><td>FMD2, FMD1, FMD0:
Timer function select bits</td><td>• These bits select the timer function mode.
• Setting the FMD2, FMD1, and FMD0 bits to "010_B" selects the PPG function mode.
• The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit3</td><td>OSEL:
Output polarity select bit</td><td>• Selects the polarity of PPG output.
<table>
<tr><th>Polarity</th><th>After reset</th><th>End of "L"-width counting</th><th>End of "H"-width counting</th></tr>
<tr><td>Normal</td><td>"L" output</td><td> </td><td> </td></tr>
<tr><td>Inverted</td><td>"H" output</td><td> </td><td> </td></tr>
</table></td></tr>
<tr><td>bit2</td><td>MDSE:
Mode select bit</td><td>• Selects continuous pulse output or one-shot pulse output.
• The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit1</td><td>CTEN:
Counting enable bit</td><td>• This bit enables the down counter.
• Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.</td></tr>
<tr><td>bit0</td><td>STRG:
Software trigger bit</td><td>• Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger.
Note:
Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.
• The value read from the STRG bit is always "0".
Note:
Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</td></tr>
</table>

■ Status Control Register (BTxSTC)

Figure 24.8-16 Status Control Register (BTxSTC)

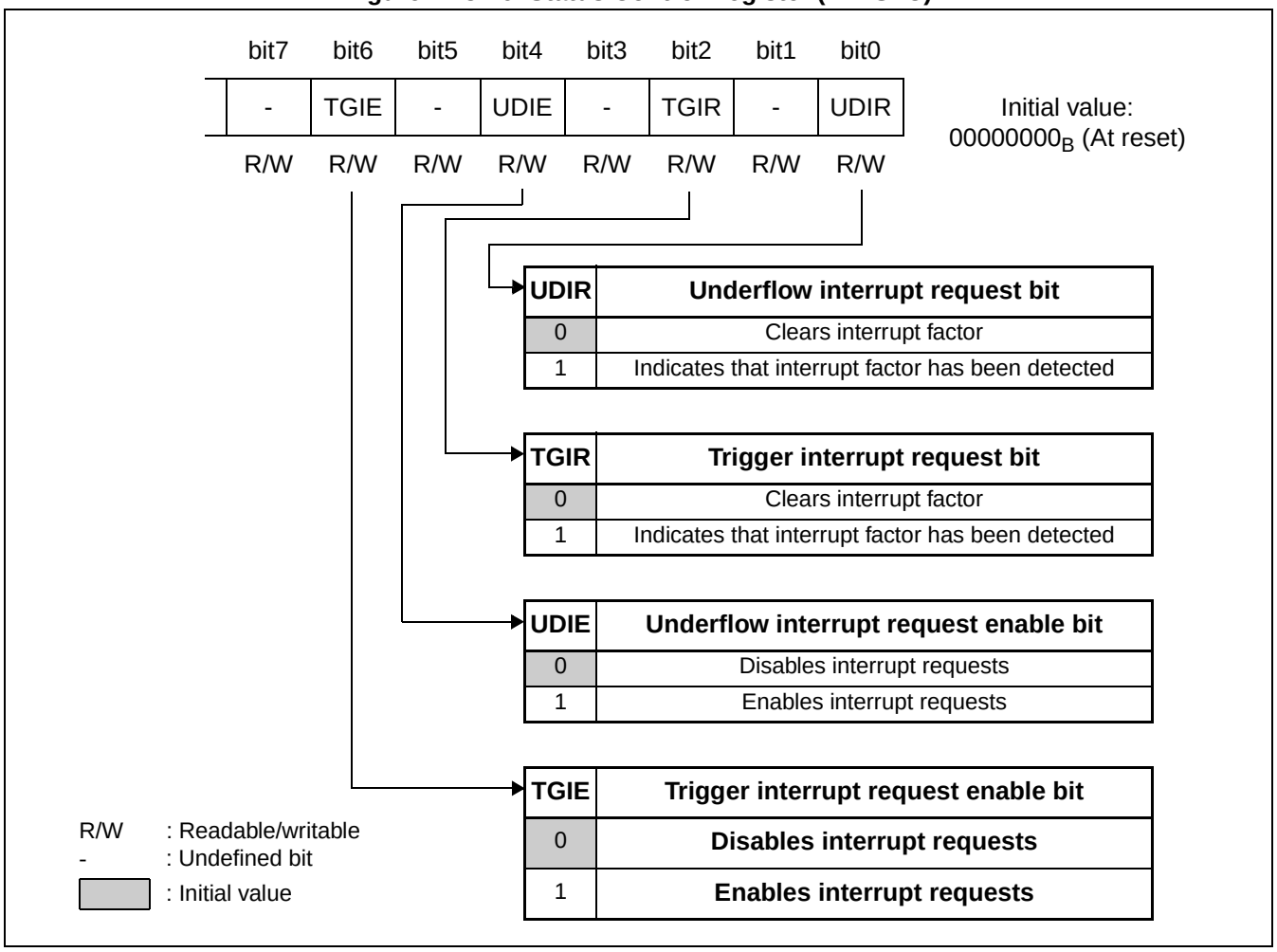

bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
- TGIE - UDIE - TGIR - UDIR
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
00000000B (At reset)
UDIR Underflow interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
TGIR Trigger interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
UDIE Underflow interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
TGIE Trigger interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-6 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	• Controls bit2: TGIR interrupt requests. • Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	• Controls bit0: UDIR interrupt requests. • Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	• The TGIR bit is set to "1" upon detection of a software trigger or trigger input. • Writing "0" to the TGIR bit clears it. • Writing "1" to the TGIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	• The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$ during counting from the value set as the "H" width. • Writing "0" to the UDIR bit clears it. • Writing "1" to the UDIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

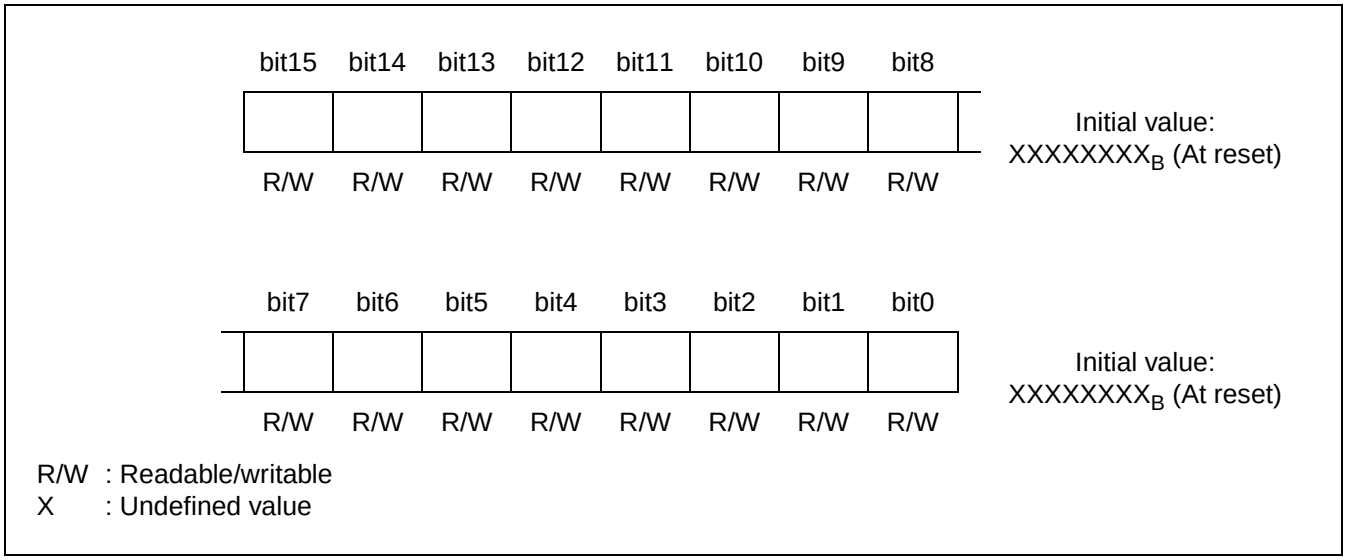
24.8.2.2 "L"-width Setting Reload Register (BTxPRLL)

The "L"-width setting reload register (BTxPRLL) is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

■ Bit Configuration of the "L"-width Setting Reload Register (BTxPRLL)

Figure 24.8-17 shows the bit configuration of the "L"-width setting reload register (BTxPRLL).

Figure 24.8-17 Bit Configuration of the "L"-width Setting Reload Register (BTxPRLL)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
R/W : Readable/writable
X : Undefined value

The BTxPRLL register is used to set the "L" width of PPG output waveforms. Transfer to the timer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting.

- Access the BTxPRLL register using 16-bit data.
- Set the "L" width using the BTxPRLL register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

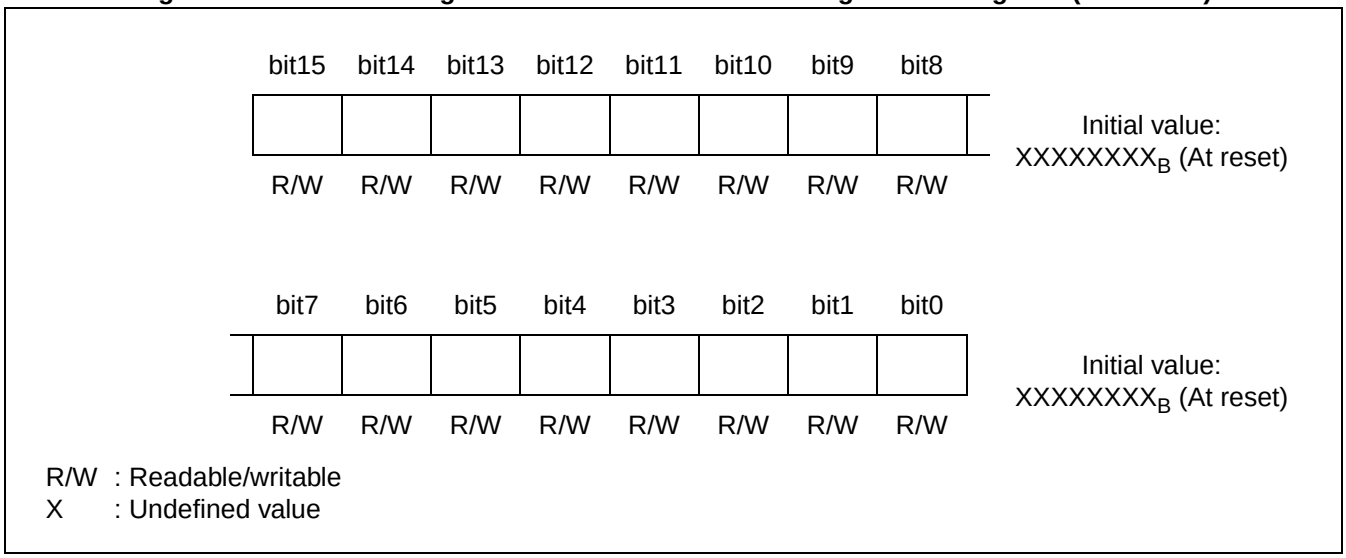
24.8.2.3 "H"-width Setting Reload Register (BTxPRLH)

The "H"-width setting reload register (BTxPRLH) is a buffered register for setting the "H" width of PPG output waveforms. Transfer from the BTxPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

■ Bit Configuration of the "H"-width Setting Reload Register (BTxPRLH)

Figure 24.8-18 shows the bit configuration of the "H"-width setting reload register (BTxPRLH).

Figure 24.8-18 Bit Configuration of the "H"-width Setting Reload Register (BTxPRLH)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
XXXXXXXX$_B$ (At reset)
R/W : Readable/writable
X : Undefined value

The BTxPRLH register is used to set the "H" width of PPG output waveforms. Transfer from the BTxPRLH register to the buffer register takes place upon detection of a start trigger or when an underflow occurs at the end of "H"-width counting. Transfer from the buffer register to the timer register takes place when an underflow occurs at the end of "L" width counting.

- Access the BTxPRLH register using 16-bit data.
- Set the "H" width using the BTxPRLH register after selecting the PPG function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.

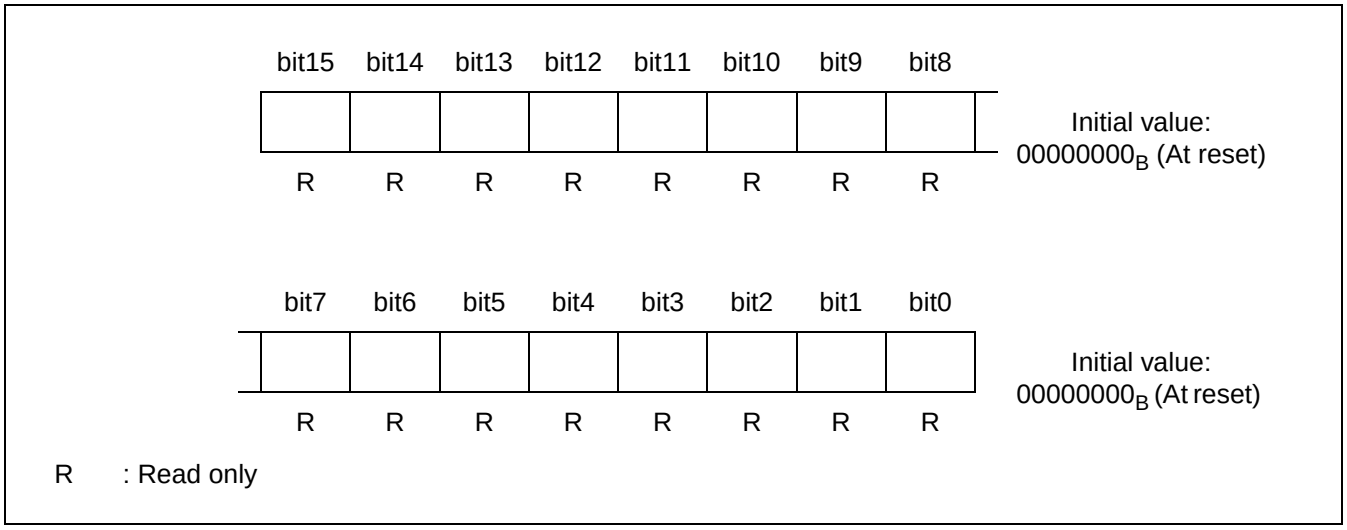
24.8.2.4 Timer Register (BTxTMR)

The timer register (BTxTMR) allows the value of the 16-bit down counter to be read from.

■ Bit Configuration of the Timer Register (BTxTMR)

Figure 24.8-19 shows the bit configuration of the PPG timer register (BTxTMR).

Figure 24.8-19 Bit Configuration of the Timer Register (BTxTMR)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
00000000B (At reset)
R R R R R R R R
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
Initial value:
00000000B (At reset)
R R R R R R R R
R : Read only

The BTxTMR register allows the value of the 16-bit down counter to be read from.

<Note>

Access the BTxTMR register using 16-bit data.

24.8.2.5 16-bit PPG Timer Operation

In PPG timer mode, an arbitrary output pulse can be controlled by setting its "L" and "H" widths in their respective reload registers.

■ Principles of Operation

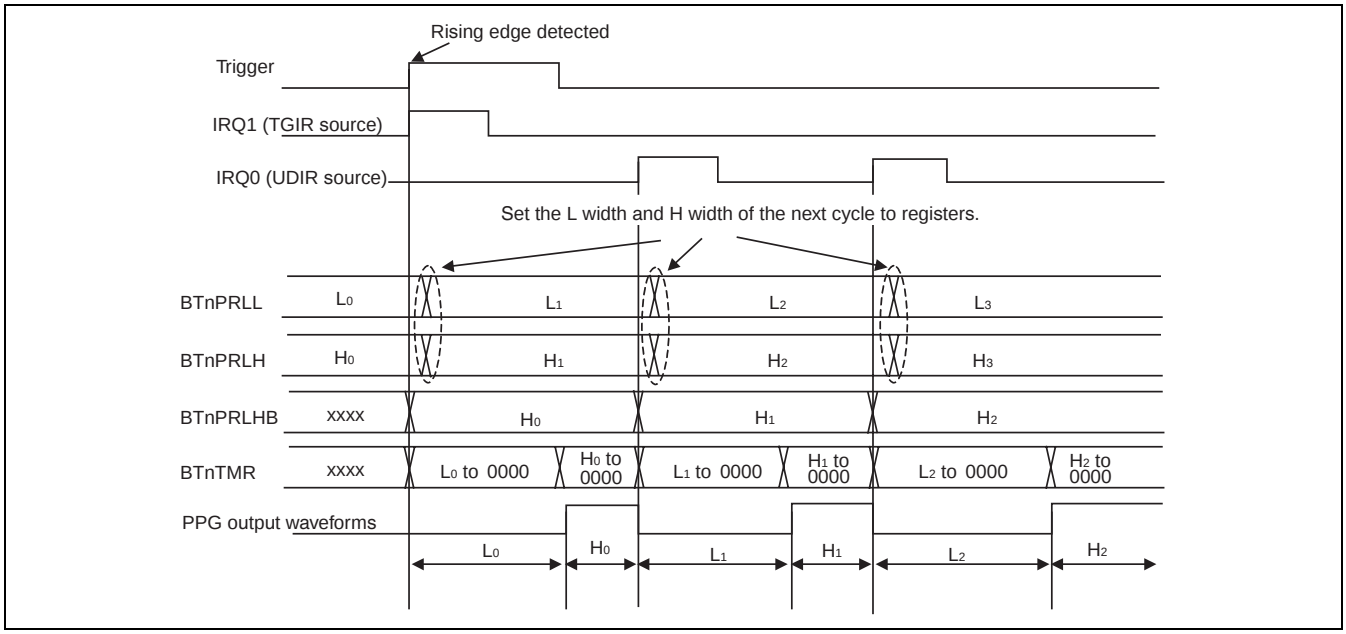
The PPG timer has two 16-bit reload registers for setting the "L" and "H" widths respectively and one "H" width setting buffer (BTxPRLL, BTxPRLH, BTxPRLHB).

In response to the start trigger, the 16-bit down counter loads the BTxPRLL value and the BTxPRLH value is transferred to the BTxPRLHB buffer at the same time. The counter is decremented every count clock with the PPG output at the "L" level. When an underflow is detected, the counter reloads the BTxPRLHB value and is decremented with the PPG output waveform inverted. When an underflow is detected again, the PPG output waveform is inverted, the counter reloads the BTxPRLL set value, and the BTxPRLH set value is transferred to the BTxPRLHB buffer.

Through these steps, the output waveform becomes the pulse output with the "L" and "H" widths corresponding to their respective reload register values.

■ Reload Register Write Timing

Data is written to the BTxPRLL and BTxPRLH reload registers upon detection of a start trigger and between when the underflow interrupt request bit (UDIR) is set and when the next period begins. The data set then becomes the setting for the next period. The BTxPRLL and BTxPRLH settings are automatically transferred to the BTxTMR and BTxPRLHB, respectively, upon detection of a start trigger and when an underflow occurs at the end of "H" width counting. The data transferred to the BTxPRLHB is automatically reloaded to the BTxTMR when an underflow occurs at the end of "L" width counting.



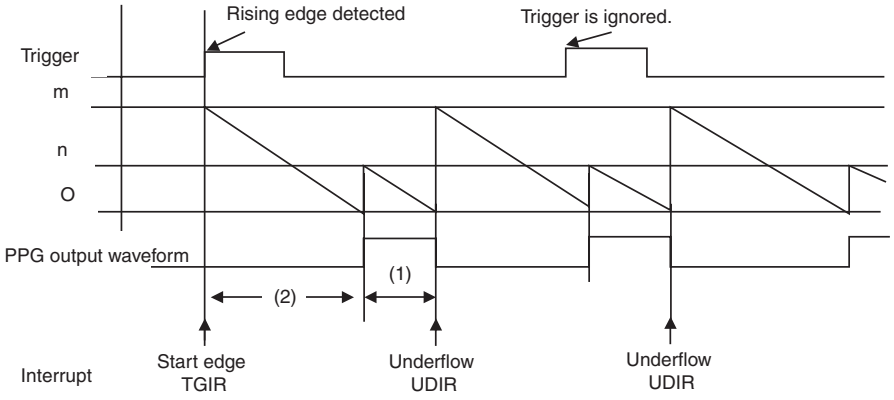
24.8.2.6 Continuous Operation

In continuous operation mode, an arbitrary pulse can be output continuously by updating the "L" and "H" widths at the set timing of each interrupt. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ Continuous Operation

● When restarting is disabled (RTGEN = 0)

Figure 24.8-20 PPG Operation Timing Chart (Restarting Disabled)

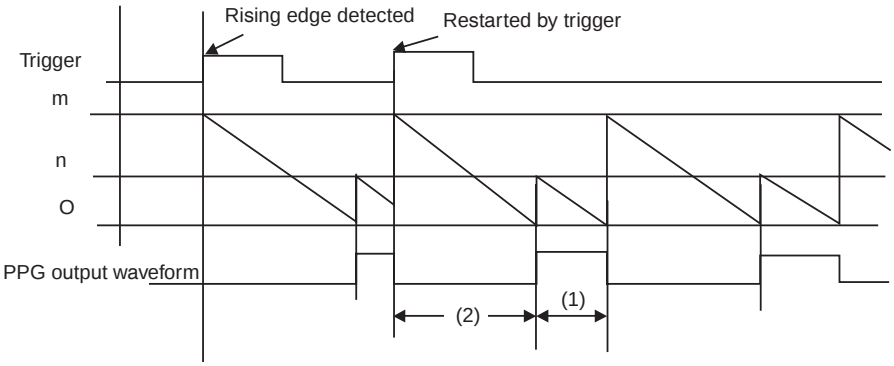


(1) = T(m+1) ms
(2) = T(n+1) ms

T : Count clock cycle
m : BTxPRLL value
n : BTxPRLH value

● When restarting is enabled (RTGEN = 1)

Figure 24.8-21 PPG Operation Timing Chart (Restarting Enabled)



(1) = T(m+1) ms
(2) = T(n+1) ms

T : Count clock cycle
m : BTxPRLL value
n : BTxPRLH value

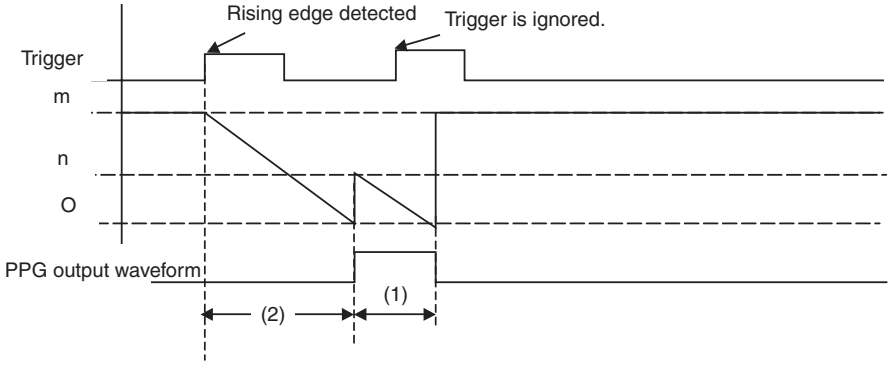
24.8.2.7 One-shot Operation

In one-shot operation mode, single pulses with an arbitrary width can be output by trigger. When restarting is enabled, the counter is reloaded upon detection of a trigger edge during operation.

■ One-shot Operation

● When restarting is disabled (RTGEN = 0)

Figure 24.8-22 One-shot Operation Timing Chart (Trigger Restarting Disabled)

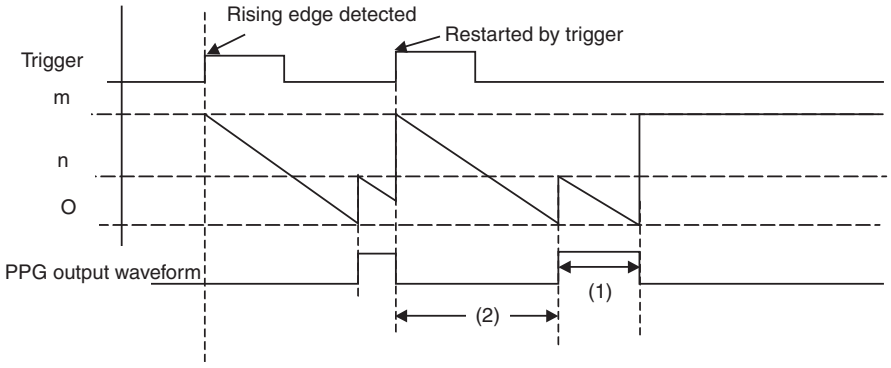


(1) = T(m+1) ms
(2) = T(n+1) ms

T : Count clock cycle
m : BTxPRLL value
n : BTxPRLH value

● When restarting is enabled (RTGEN = 1)

Figure 24.8-23 One-shot Operation Timing Chart (Trigger Restarting Enabled)



(1) = T(m+1) ms
(2) = T(n+1) ms

T : Count clock cycle
m : BTxPRLL value
n : BTxPRLH value

■ Relationship between Reload Value and Pulse Width

The output pulse width is obtained by adding 1 to the value written in the 16-bit reload register and multiplying the result by the count clock cycle. When the reload register value is 0000_H, therefore, the output has a pulse width of one count clock cycle. When the reload register value is $FFFF_H$, the output has a pulse width of 65536 count clock cycles. The pulse width is calculated from the following equation.

$PL = T \times (L+1)$

$PH = T \times (H+1)$

PL : "L" pulse width
PH : "H" pulse width
T : Count clock cycle
L : BTxPRLL value
H : BTxPRLH value

24.8.2.8 Interrupt Factors and Timing Chart

This section provides the interrupt factors and timing chart.

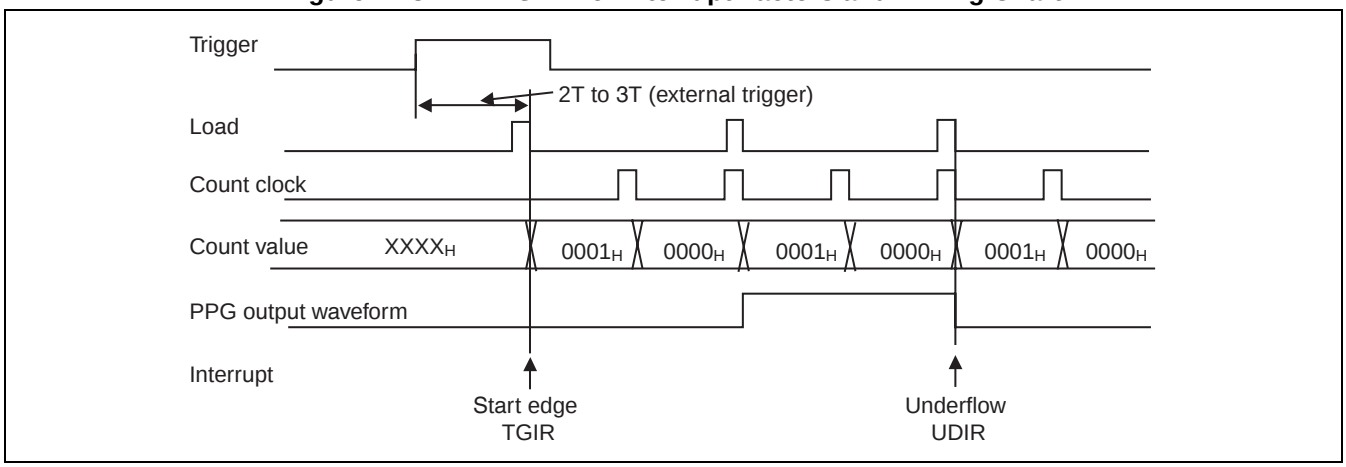
■ Interrupt Factors and Timing Chart (PPG Output: Normal Polarity)

A software trigger requires T and an external trigger requires 2T to 3T (T: peripheral clock (PCLK) cycle) until the counter value is loaded after the trigger is generated.

Interrupt factors are set when the PPG start trigger is detected and when an underflow is detected during "H" level output.

Figure 24.8-24 shows the interrupt factors and timing chart, assuming "L" width setting = 1 and "H" width setting = 1.

Figure 24.8-24 PPG Timer Interrupt Factors and Timing Chart



24.8.3 Reload Timer Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the reload timer.

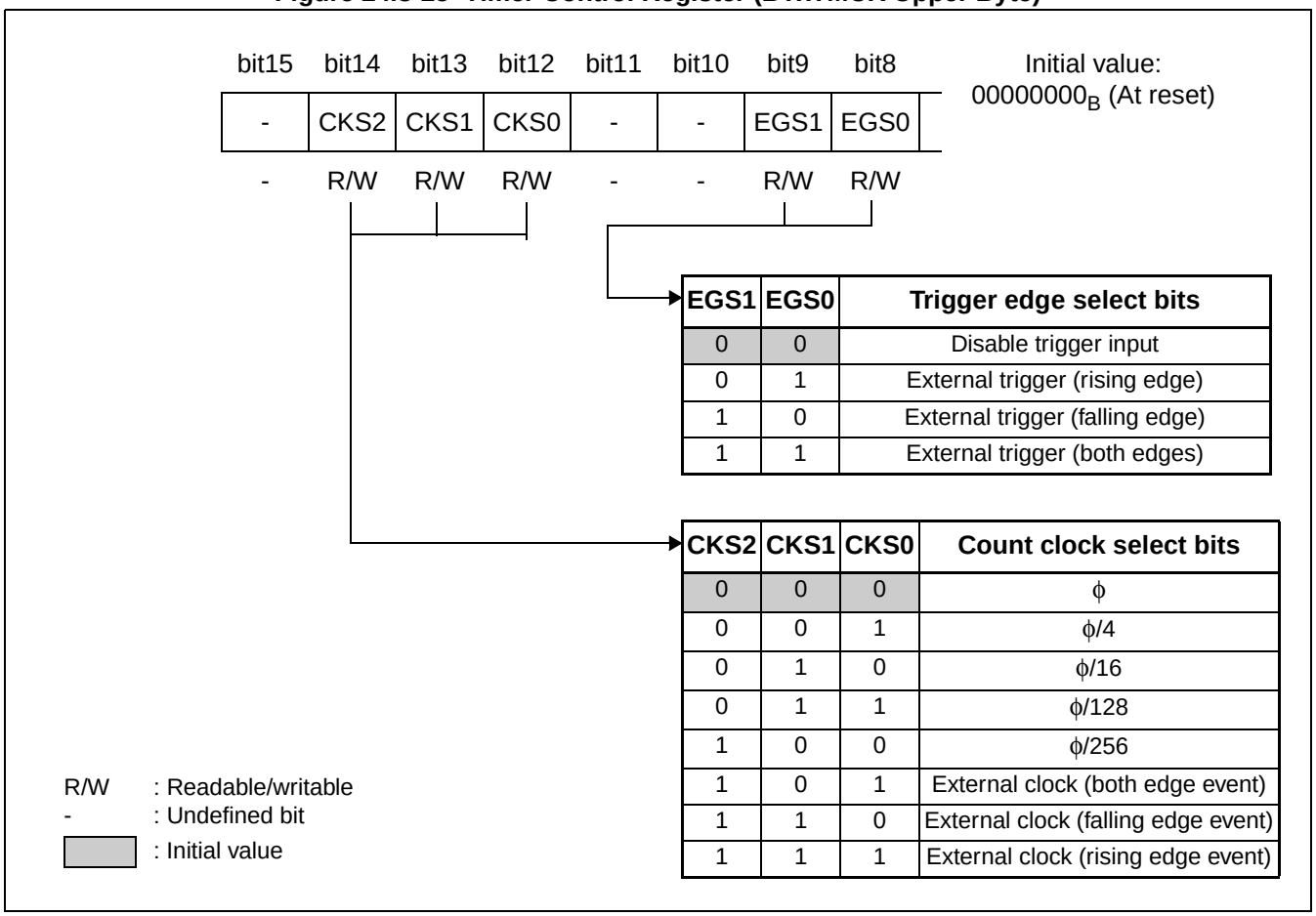
- Timer Control Register (BTxTMCR) for Reload Timer
- Period Setting Register (BTxPCSR)
- Timer Register (BTxTMR)
- 16-bit Reload Timer Operation

24.8.3.1 Timer Control Register (BTxTMCR) for Reload Timer

The timer control register (BTxTMCR) controls the reload timer.

■ Timer Control Register (BTxTMCR Upper Byte)

Figure 24.8-25 Timer Control Register (BTxTMCR Upper Byte)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
00000000B (At reset)
- CKS2 CKS1 CKS0 - - EGS1 EGS0
- R/W R/W R/W - - R/W R/W
R/W : Readable/writable
- : Undefined bit
: Initial value

EGS1	EGS0	Trigger edge select bits
0	0	Disable trigger input
0	1	External trigger (rising edge)
1	0	External trigger (falling edge)
1	1	External trigger (both edges)

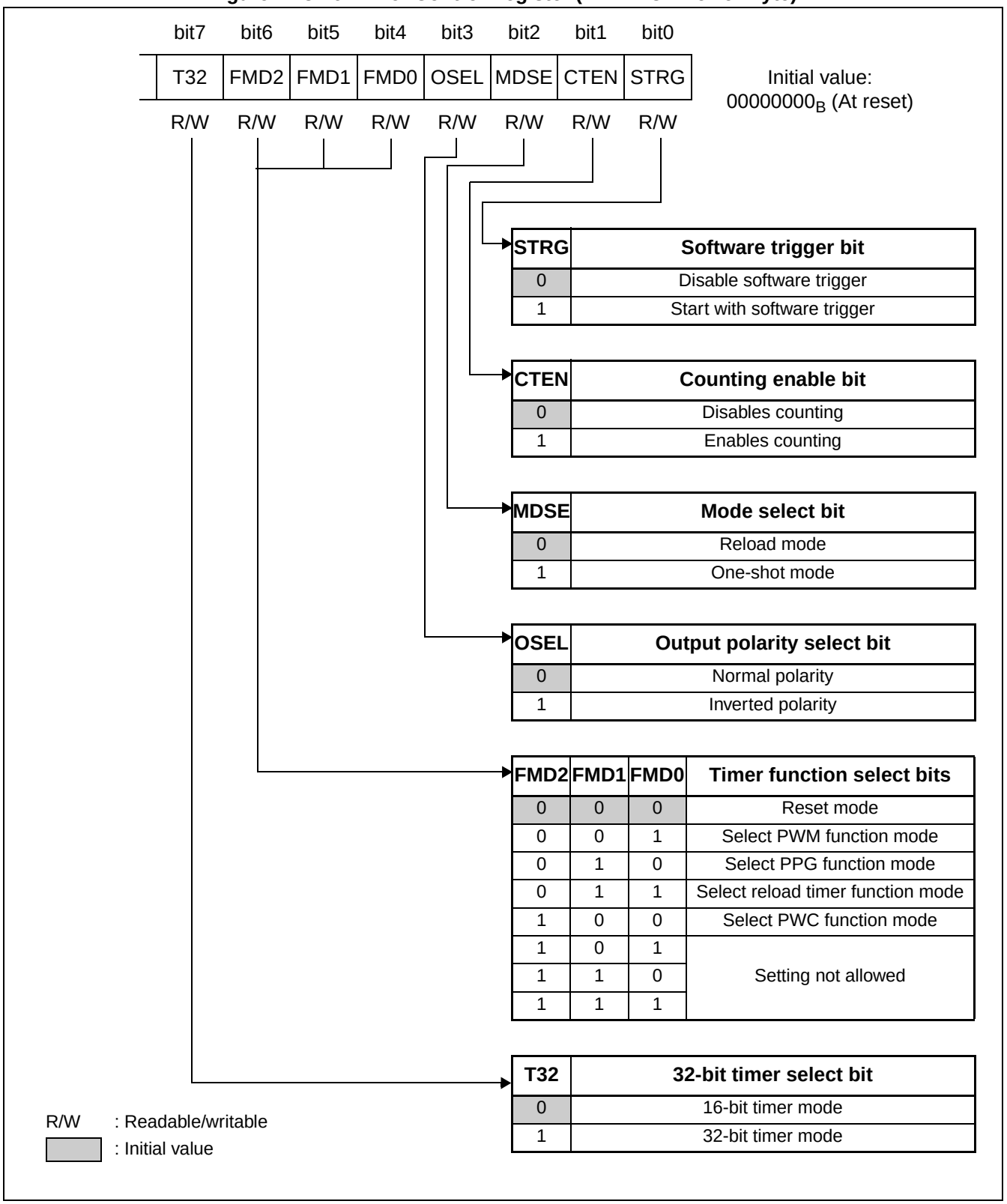
CKS2	CKS1	CKS0	Count clock select bits
0	0	0	ϕ
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	External clock (both edge event)
1	1	0	External clock (falling edge event)
1	1	1	External clock (rising edge event)

Table 24.8-7 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	• The read value of this bit is undefined. • Write to this bit takes no effect.
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	• Select the count clock for the 16-bit down counter. • The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11, bit10	Undefined bits	• The value read is "0" • When writing to these bits, write "0".
bit9, bit8	EGS1, EGS0: Trigger edge select bits	• Select the effective edge of the input waveform as an external trigger to set the trigger condition. • When these bits are set to the initial value or "00_B", no effective edge of the input waveform is selected, preventing the timer from being triggered by the external waveform. Note: Writing "1" to the STRG bit enables the software trigger irrespective of the settings of EGS1 and EGS0. • EGS1 and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTxTMCR Lower Byte)

Figure 24.8-26 Timer Control Register (BTxTMCR Lower Byte)


bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
T32 FMD2 FMD1 FMD0 OSEL MDSE CTEN STRG
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
00000000B (At reset)
R/W : Readable/writable
: Initial value

STRG	Software trigger bit
0	Disable software trigger
1	Start with software trigger

CTEN	Counting enable bit
0	Disables counting
1	Enables counting

MDSE	Mode select bit
0	Reload mode
1	One-shot mode

OSEL	Output polarity select bit
0	Normal polarity
1	Inverted polarity

FMD2	FMD1	FMD0	Timer function select bits
0	0	0	Reset mode
0	0	1	Select PWM function mode
0	1	0	Select PPG function mode
0	1	1	Select reload timer function mode
1	0	0	Select PWC function mode
1	0	1	Setting not allowed
1	1	0	
1	1	1	

T32	32-bit timer select bit
0	16-bit timer mode
1	32-bit timer mode

Table 24.8-8 Timer Control Register (BTxTMCR Lower Byte)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit7</td><td>T32:
32-bit timer select bit</td><td>• This bit selects the 32-bit timer mode.
• When the FMD2, FMD1, and FMD0 bits contain "011_B" to select the reload timer, setting the T32 bit to "1" places the timer in 32-bit timer mode.
• The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
→ See Section "24.5 32-bit Mode Operations".</td></tr>
<tr><td>bit6
to
bit4</td><td>FMD2, FMD1, FMD0:
Timer function select bits</td><td>• These bits select the timer function mode.
• Setting the FMD2, FMD1, and FMD0 bits to "011_B" selects the reload timer function mode.
• The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit3</td><td>OSEL:
Output polarity select bit</td><td>• Selects the timer output at normal level or inverted level.
• The output waveform is generated as follows depending on the combination with the MDSE bit (bit2):
<table>
<tr><th>MDSE</th><th>OSEL</th><th>Output Waveforms</th></tr>
<tr><td>0</td><td>0</td><td>Toggle output of "L" at the count start</td></tr>
<tr><td>0</td><td>1</td><td>Toggle output of "H" at the count start</td></tr>
<tr><td>1</td><td>0</td><td>Rectangular wave of "H" during count</td></tr>
<tr><td>1</td><td>1</td><td>Rectangular wave of "L" during count</td></tr>
</table></td></tr>
<tr><td>bit2</td><td>MDSE:
Mode select bit</td><td>• Setting the MDSE bit to "0" selects reload mode, in which the counter loads the reload register value to continue counting the moment a count value underflow occurs from 0000_H to $FFFF_H$.
• Setting the MDSE bit to "1" selects one-shot mode, in which the counter stops operation the moment a count value underflow occurs from 0000_H to $FFFF_H$.
• The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.</td></tr>
<tr><td>bit1</td><td>CTEN:
Counting enable bit</td><td>• This bit enables the down counter.
• Writing "0" to the CTEN bit with the counter enabled (CTEN = 1) stops the counter.</td></tr>
<tr><td>bit0</td><td>STRG:
Software trigger bit</td><td>• Writing "1" to the STRG bit with the CTEN bit containing "1" generates a software trigger.
Note:
Writing "1" to the CTEN and STRG bits at the same time also generates a software trigger.
• The value read from the STRG bit is always "0".
Note:
Writing "1" to the STRG bit enables the software trigger irrespective of the settings of the EGS1 and EGS0 bits.</td></tr>
</table>

■ Status Control Register (BTxSTC)

Figure 24.8-27 Status Control Register (BTxSTC)

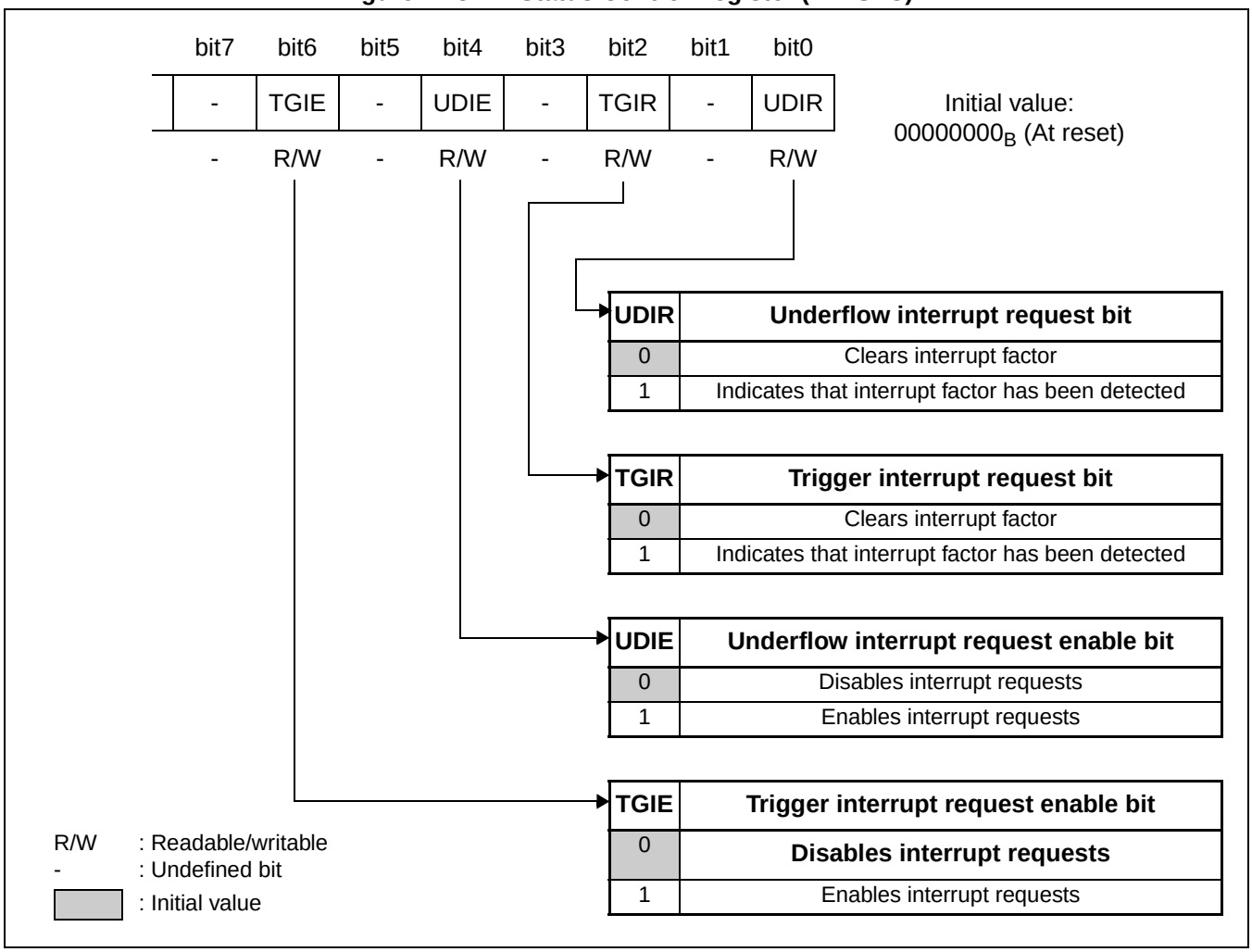

bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
- TGIE - UDIE - TGIR - UDIR
- R/W - R/W - R/W - R/W
Initial value:
00000000B (At reset)
UDIR Underflow interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
TGIR Trigger interrupt request bit
0 Clears interrupt factor
1 Indicates that interrupt factor has been detected
UDIE Underflow interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
TGIE Trigger interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-9 Status Control Register (BTxSTC)

Bit name		Function
bit7	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit6	TGIE: Trigger interrupt request enable bit	• Controls bit2:TGIR interrupt requests. • Setting the TGIR bit (bit2) with the TGIE bit enabling trigger interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit4	UDIE: Underflow interrupt request enable bit	• Controls bit0:UDIR interrupt requests. • Setting the UDIR bit (bit0) with the UDIE bit enabling underflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit2	TGIR: Trigger interrupt request bit	• The TGIR bit is set to "1" upon detection of a software trigger or trigger input. • Writing "0" to the TGIR bit clears it. • Writing "1" to the TGIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.
bit1	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit0	UDIR: Underflow interrupt request bit	• The UDIR bit is set to "1" when a count value underflow occurs from 0000_H to $FFFF_H$. • Writing "0" to the UDIR bit clears it. • Writing "1" to the UDIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

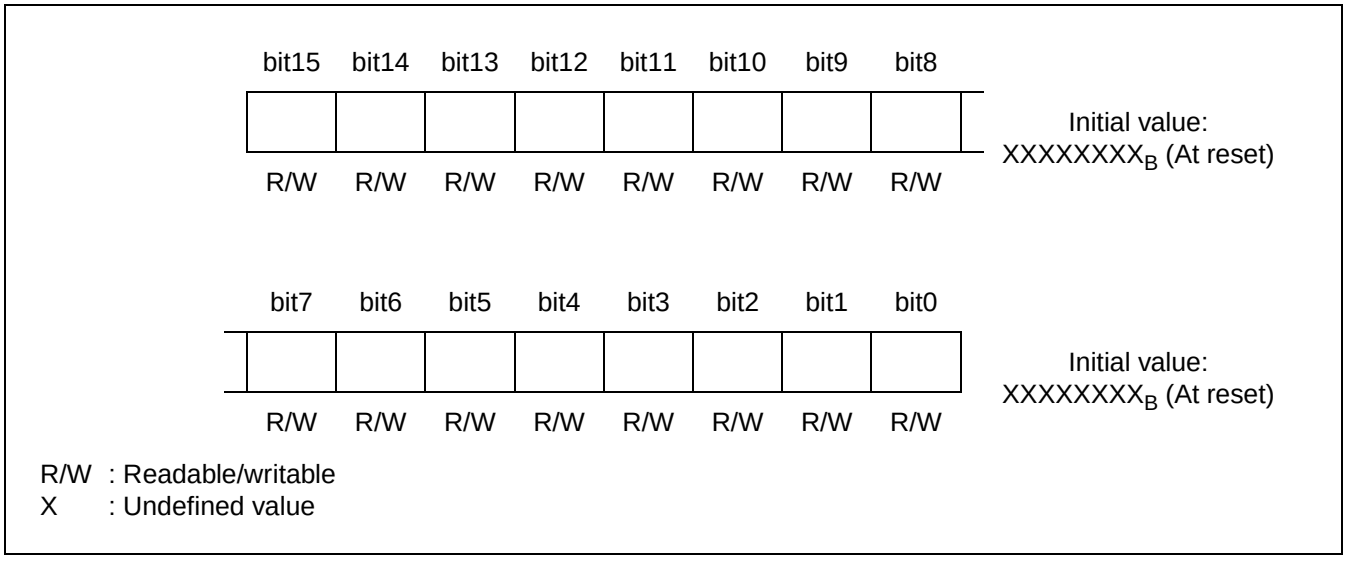
24.8.3.2 Period Setting Register (BTxPCSR)

The period setting register (BTxPCSR) holds the initial count value. In 32-bit mode, the register holds the initial count value of the lower 16 bits for the even-numbered channel or the initial count value of the upper 16 bits for the odd-numbered channel. The initial value immediately after a reset is undefined. To access this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Period Setting Register (BTxPCSR)

Figure 24.8-28 shows the bit configuration of the period setting register (BTxPCSR).

Figure 24.8-28 Bit Configuration of the Period Setting Register (BTxPCSR)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
XXXXXXXX$_B$ (At reset)
R/W R/W R/W R/W R/W R/W R/W R/W
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
Initial value:
XXXXXXXX$_B$ (At reset)
R/W R/W R/W R/W R/W R/W R/W R/W
R/W : Readable/writable
X : Undefined value

The BTxPCSR register is used to set the period. Transfer to the timer register takes place when an underflow occurs.

- Access the BTxPCSR register using 16-bit data.
- Set the period using the BTxPCSR register after selecting the reload timer function mode using the FMD2, FMD1, and FMD0 bits in the BTxTMCR register.
- To write data to the BTxPCSR register in 32-bit mode, access its upper 16-bit data (data for the odd-numbered channel) first and then the lower 16-bit data (data for the even-numbered channel).

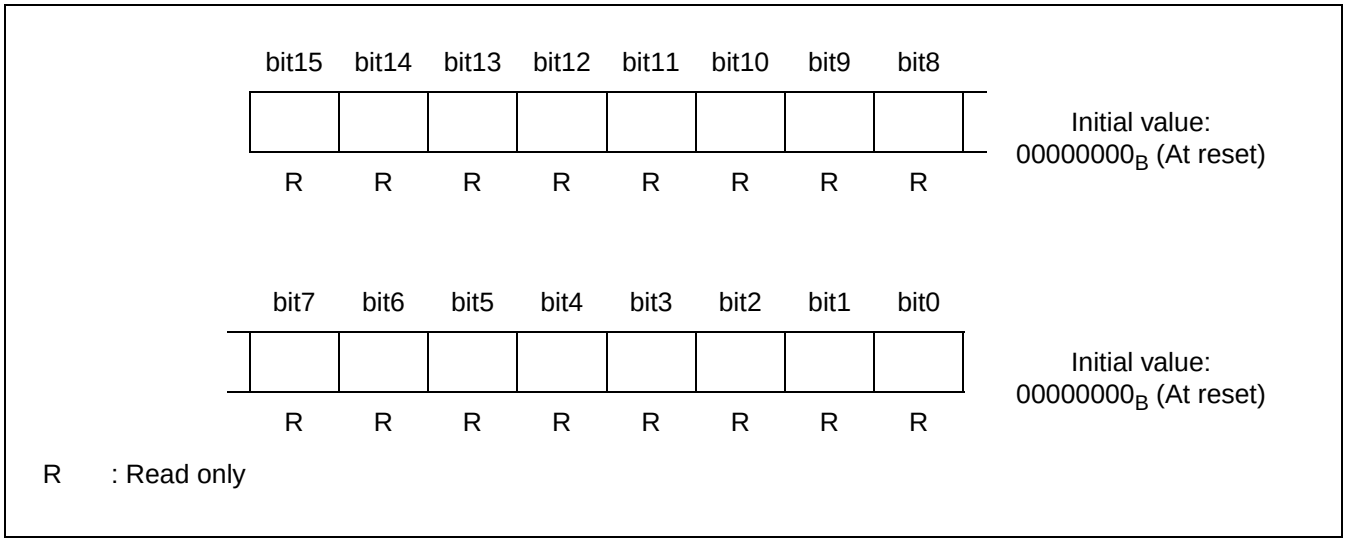
24.8.3.3 Timer Register (BTxTMR)

The timer register (BTxTMR) allows the count value of the timer to be read from. In 32-bit mode, the register holds the count value of the lower 16 bits for the even-numbered channel or the count value for the upper 16 bits for the odd-numbered channel. The initial value is undefined.
To read this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Timer Register (BTxTMR)

Figure 24.8-29 shows the bit configuration of the timer register (BTxTMR).

Figure 24.8-29 Bit Configuration of the Timer Register (BTxTMR)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
R R R R R R R R
Initial value:
00000000B (At reset)
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
R R R R R R R R
Initial value:
00000000B (At reset)
R : Read only

The BTxTMR register allows the value of the 16-bit down counter to be read from.

<Notes>

- Access the BTxTMR register using 16-bit data.
- To read data from the BTxTMR register in 32-bit mode, access its lower 16-bit data (data for the even-numbered channel) first and then the upper 16-bit data (data for the odd-numbered channel).

24.8.3.4 16-bit Reload Timer Operation

In reload timer mode, the timer decrements the counter from the value set in the period setting register in synchronization with the count clock, and finishes counting when the count value reaches "0" or continues operation with the period setting loaded automatically until the counter stops being decremented.

■ Counting with the Internal Clock Selected

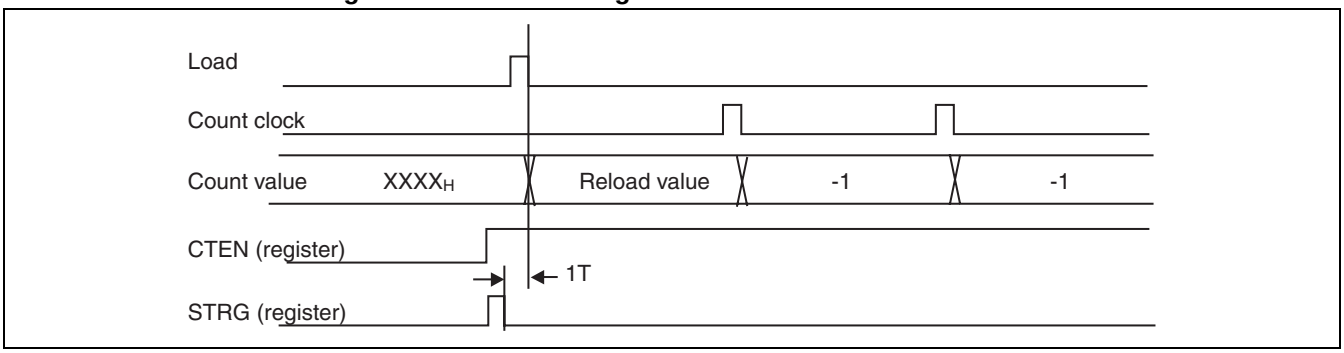
To start counting the moment counting is enabled, write "1" to both of the CTEN and STRG bits in the timer control register. The STRG bit maintains the trigger input always enabled irrespective of the operation mode as long as the timer is active (CNTE = 1).

Enable counting and start the timer using a software trigger or external trigger, and the timer loads the period setting register value to the counter to start decrementing the counter.

It takes 1T (T: peripheral clock (PCLK) cycle) for data in the period setting register to be loaded into the counter after the counter start trigger is set.

Figure 24.8-30 illustrates how the counter is started by the software trigger and operates.

Figure 24.8-30 Counting with the Internal Clock Selected



■ Underflow Operation

When the counter value changes from "0000_H" to "$FFFF_H$", the transition is detected as an underflow.

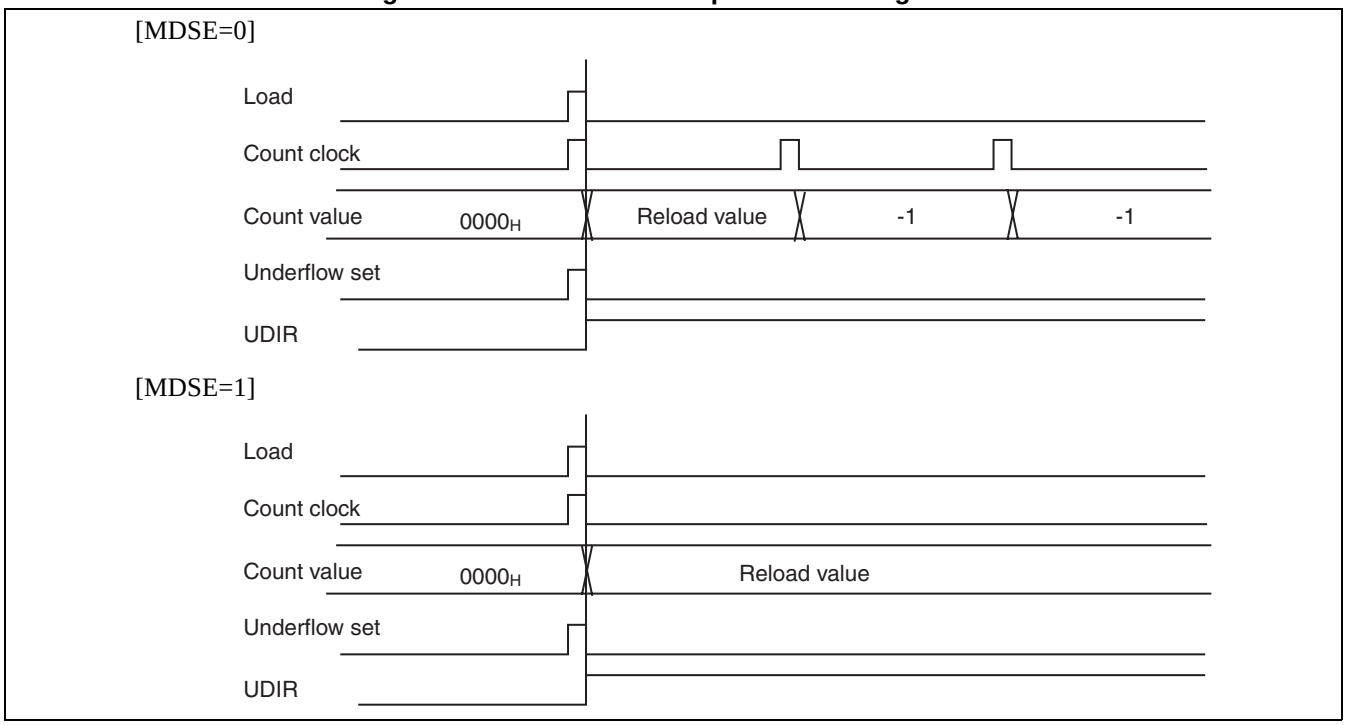
When the counter counts [period setting register value + 1], therefore, an underflow occurs.

When an underflow occurs, the content of the period setting register (BTxPCSR) is loaded into the counter, and the counter continues counting if the MDSE bit in the timer control register (BTxTMCR) is "0". If the MDSE bit is "1", the counter stops operation with the loaded counter value left unchanged.

When an underflow occurs, the UDIR bit in the status control register (BTxSTC) is set and an interrupt request occurs if the UDIE bit is "1".

Figure 24.8-31 is a timing chart of underflow operation.

Figure 24.8-31 Underflow Operation Timing Chart

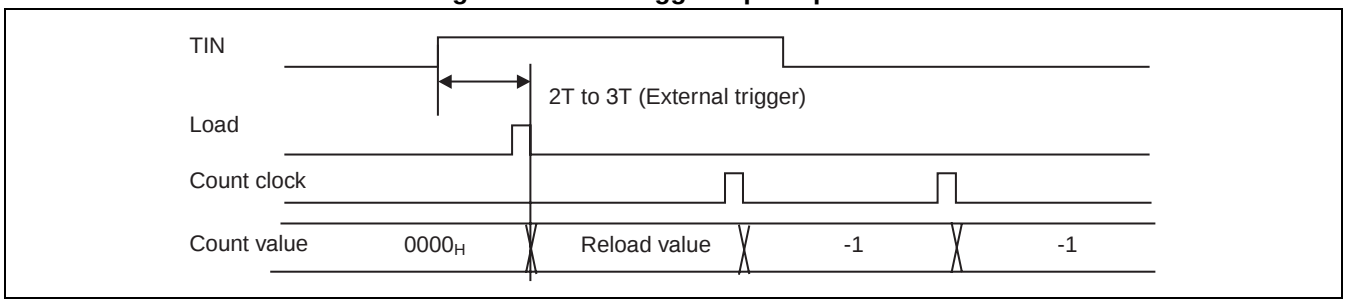


■ Input Pin Operation

The TIN pin can be used as a trigger input. When the effective edge is input to the TIN pin, the counter loads the content of the period setting register and starts counting. It takes 2T or 3T (T: peripheral clock (PCLK) cycle) for the counter value to be loaded after the trigger is applied.

Figure 24.8-32 illustrates the trigger input operation with the rising edge selected as the effective edge.

Figure 24.8-32 Trigger Input Operation

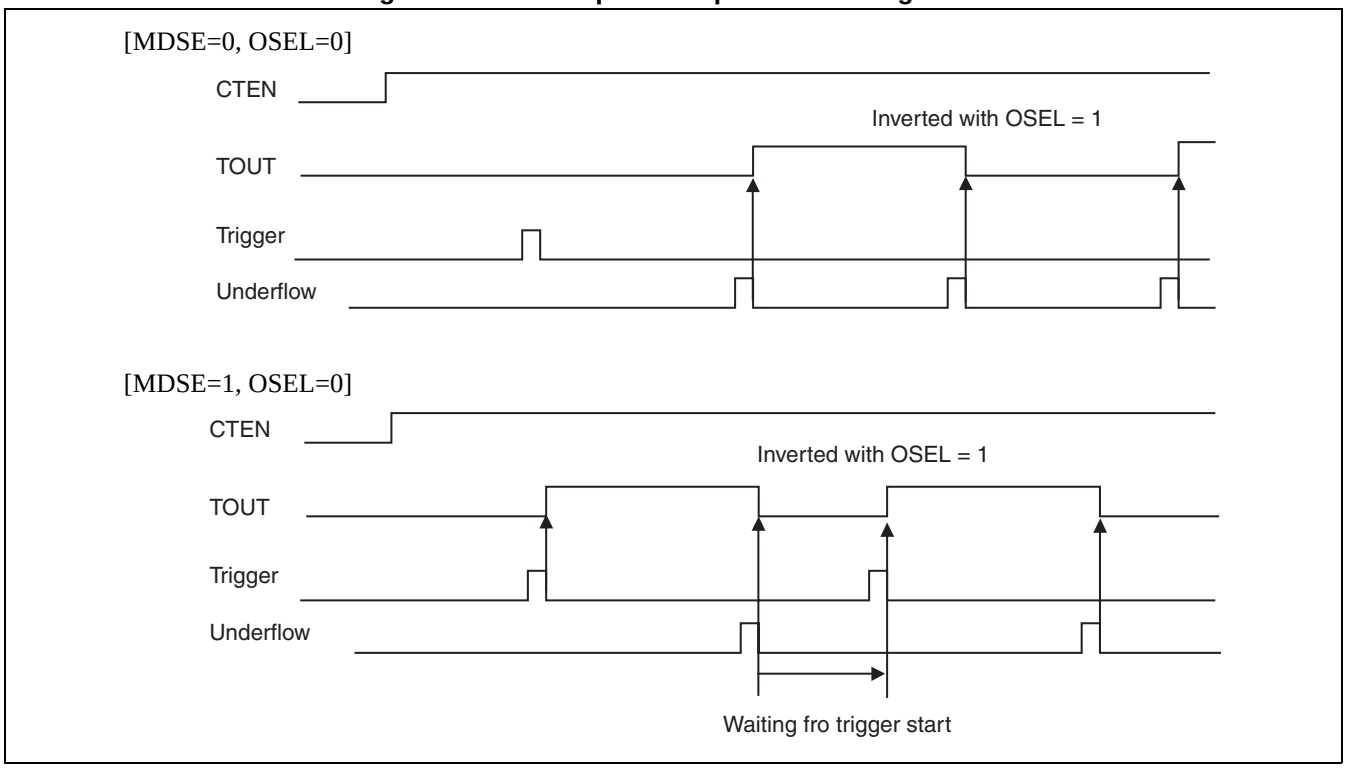


■ Output Pin Operation

The TOUT pin functions as a toggle output to be inverted at each underflow in reload mode and as a pulse output to indicate that counting is in process in one-shot mode. The output polarity can be set by the OSEL bit in the timer control register (BTxTMCR). When the OSEL bit is "0", the initial value of the toggle output is "0" and that of the one-shot pulse output is "1" (indicating that counting is in process). Setting the OSEL bit to "1" inverts the output waveform.

Figure 24.8-33 is a timing chart of output pin operation.

Figure 24.8-33 Output Pin Operation Timing Chart



24.8.4 PWC Function

The base timer can assign itself, according to the settings of the FMD2, FMD1, and FMD0 bits in its timer control register, to serve as only one of the 16-bit PWM timer, 16-bit PPG timer, 16/32-bit reload timer, and 16/32-bit PWC timer. This section describes the functions of the base timer assigned as the PWC timer.

- Timer Control Register (BTxTMCR) for PWC Timer
- Data Buffer Register (BTxDTBF)
- PWC Operation

24.8.4.1 Timer Control Register (BTxTMCR) for PWC Timer

The timer control register (BTxTMCR) controls the PWC timer.

■ Timer Control Register (BTxTMCR Upper Byte)

Figure 24.8-34 Timer Control Register (BTxTMCR Upper Byte)

bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
-	CKS2	CKS1	CKS0	-	EGS2	EGS1	EGS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Initial value: 00000000_B (At reset)

EGS2	EGS1	EGS0	Measurement edge select bits
0	0	0	Measure "H" pulse width (↑ to ↓)
0	0	1	Measure period between rising edges (↑ to ↑)
0	1	0	Measure period between falling edges (↓ to ↓)
0	1	1	Measure pulse widths between all edges (↑ or ↓ to ↓ or ↑)
1	0	0	Measure "L" pulse width (↓ to ↑)
1	0	1	Setting not allowed
1	1	0	
1	1	1	

CKS2	CKS1	CKS0	Count clock select bits
0	0	0	ϕ
0	0	1	ϕ/4
0	1	0	ϕ/16
0	1	1	ϕ/128
1	0	0	ϕ/256
1	0	1	**Setting not allowed**
1	1	0	
1	1	1	

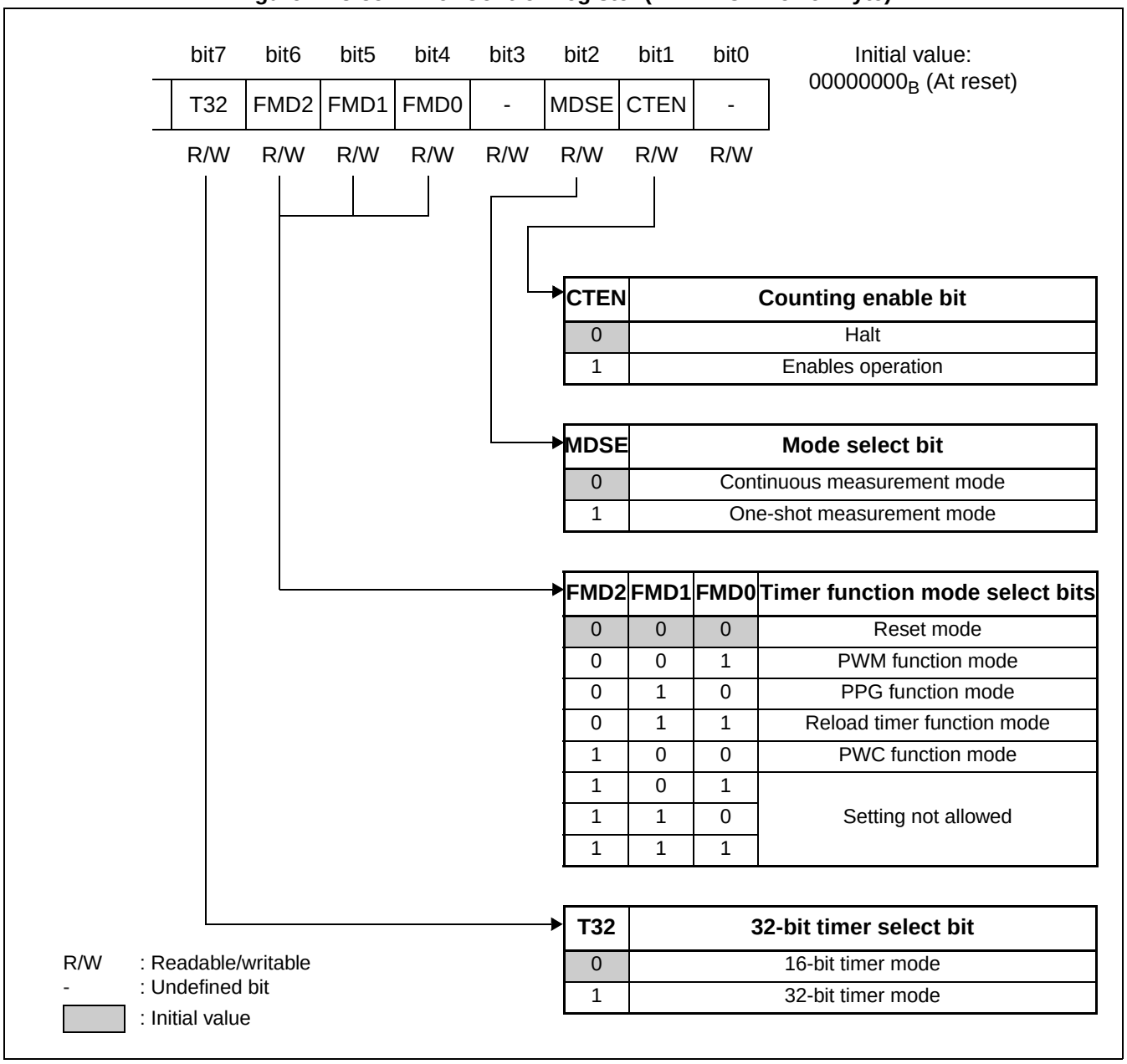
R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-10 Timer Control Register (BTxTMCR Upper Byte)

Bit name		Function
bit15	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit14 to bit12	CKS2, CKS1, CKS0: Count clock select bits	• Select the count clock for the 16-bit up counter. • The count clock promptly reflects any changes made to its setting. CKS2 to CKS0 must therefore be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.
bit11	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit10 to bit8	EGS2, EGS1, EGS0: Measurement edge select bits	• Set the measurement edge condition. • EGS2, EGS1, and EGS0 must be updated while counting is stopped (CTEN = 0). Note, however, that you can change their setting at the same time as writing "1" to the CTEN bit.

■ Timer Control Register (BTxTMCR Lower Byte)

Figure 24.8-35 Timer Control Register (BTxTMCR Lower Byte)


bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
T32 FMD2 FMD1 FMD0 - MDSE CTEN -
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
00000000B (At reset)
R/W : Readable/writable
- : Undefined bit
: Initial value

CTEN	Counting enable bit
0	Halt
1	Enables operation

MDSE	Mode select bit
0	Continuous measurement mode
1	One-shot measurement mode

FMD2	FMD1	FMD0	Timer function mode select bits
0	0	0	Reset mode
0	0	1	PWM function mode
0	1	0	PPG function mode
0	1	1	Reload timer function mode
1	0	0	PWC function mode
1	0	1	Setting not allowed
1	1	0	
1	1	1	

T32	32-bit timer select bit
0	16-bit timer mode
1	32-bit timer mode

Table 24.8-11 Timer Control Register (BTxTMCR Lower Byte)

Bit name		Function
bit7	T32: 32-bit timer select bit	• This bit selects the 32-bit timer mode. • When the FMD2, FMD1, and FMD0 bits contain "100_B" to select the PWC timer, setting the T32 bit to "1" places the timer in 32-bit PWC mode. • The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit. → See Section "24.5 32-bit Mode Operations".
bit6 to bit4	FMD2, FMD1, FMD0: Timer function mode select bits	• These bits select the timer function mode. • Setting the FMD2, FMD1, and FMD0 bits to "100_B" selects the PWC timer function mode. • The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit3	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit2	MDSE: Mode select bit	• Selects measurement mode as follows. **MDSE** / **Mode** / **Operation** 0 / Continuous measurement / Continuous measurement: buffer register enabled 1 / One-shot measurement / Halts after each measurement • The setting must be changed with the timer stopped (CTEN = 0). Note, however, that you can change the setting at the same time as writing "1" to the CTEN bit.
bit1	CTEN: Counting enable bit	• This bit enables the starting or restarting of the up counter. • Writing "1" to this bit with the counter enabled for operation (CTEN bit = 1) causes a restart, resulting in the counter cleared and waiting for the measurement start edge. • Writing "0" to the bit with the counter enabled for operation (CTEN bit = 1 stops the counter.
bit0	Undefined bit	• The value read is "0" • When writing to this bit, write "0".

■ Status Control Register (BTxSTC)

Figure 24.8-36 Status Control Register (BTxSTC)

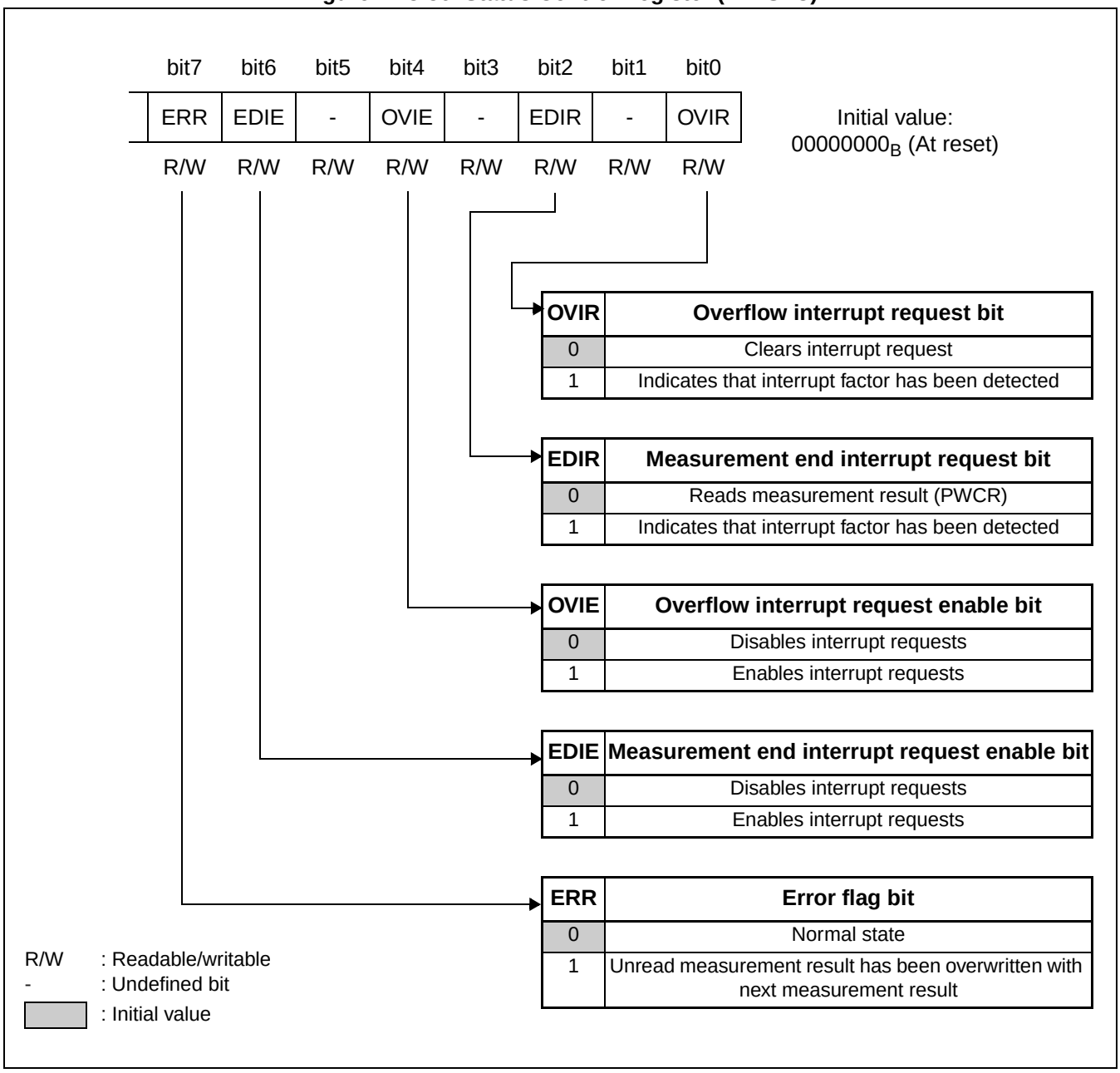

bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
ERR EDIE - OVIE - EDIR - OVIR
R/W R/W R/W R/W R/W R/W R/W R/W
Initial value:
00000000B (At reset)
OVIR Overflow interrupt request bit
0 Clears interrupt request
1 Indicates that interrupt factor has been detected
EDIR Measurement end interrupt request bit
0 Reads measurement result (PWCR)
1 Indicates that interrupt factor has been detected
OVIE Overflow interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
EDIE Measurement end interrupt request enable bit
0 Disables interrupt requests
1 Enables interrupt requests
ERR Error flag bit
0 Normal state
1 Unread measurement result has been overwritten with next measurement result
R/W : Readable/writable
- : Undefined bit
: Initial value

Table 24.8-12 Status Control Register (BTxSTC)

Bit name		Function
bit7	ERR: Error flag bit	• This flag indicates that the next measurement has been completed before reading the current measurement result from the BTxDTBF register in continuous measurement mode. In this case, the BTxDTBF register is updated with the new measurement result, discarding the preceding measurement result. • Measurement continues irrespective of the ERR bit value. • The ERR bit can only be read; an attempt to write to it has no effect on the bit value. • The ERR bit is cleared by reading the measurement result (BTxDTBF).
bit6	EDIE: Measurement end interrupt request enable bit	• Controls bit2: EDIR interrupt requests. • Setting the EDIR bit (bit2) with the EDIE bit enabling measurement end interrupt requests generates an interrupt request to the CPU.
bit5	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit4	OVIE: Overflow interrupt request enable bit	• Controls bit0: OVIR interrupt requests. • Setting the OVIR bit (bit0) with the OVIE bit enabling overflow interrupt requests generates an interrupt request to the CPU.
bit3	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit2	EDIR: Measurement end interrupt request bit	• Indicates that measurement has been completed. The flag is set to "1" upon completion. • The EDIR bit is cleared by reading the measurement result (BTxDTBF). • The EDIR bit can only be read; an attempt to write to it has no effect on the bit value.
bit1	Undefined bit	• The value read is "0" • When writing to this bit, write "0".
bit0	OVIR: Overflow interrupt request bit	• The flag is set to "1" when a count value overflow occurs from $FFFF_H$ to 0000_H. • Writing "0" to the OVIR bit clears it. • Writing "1" to the OVIR bit has no effect on the bit value. • When read by a read modify write (RMW) instruction, the bit always returns "1" irrespective of the current bit value.

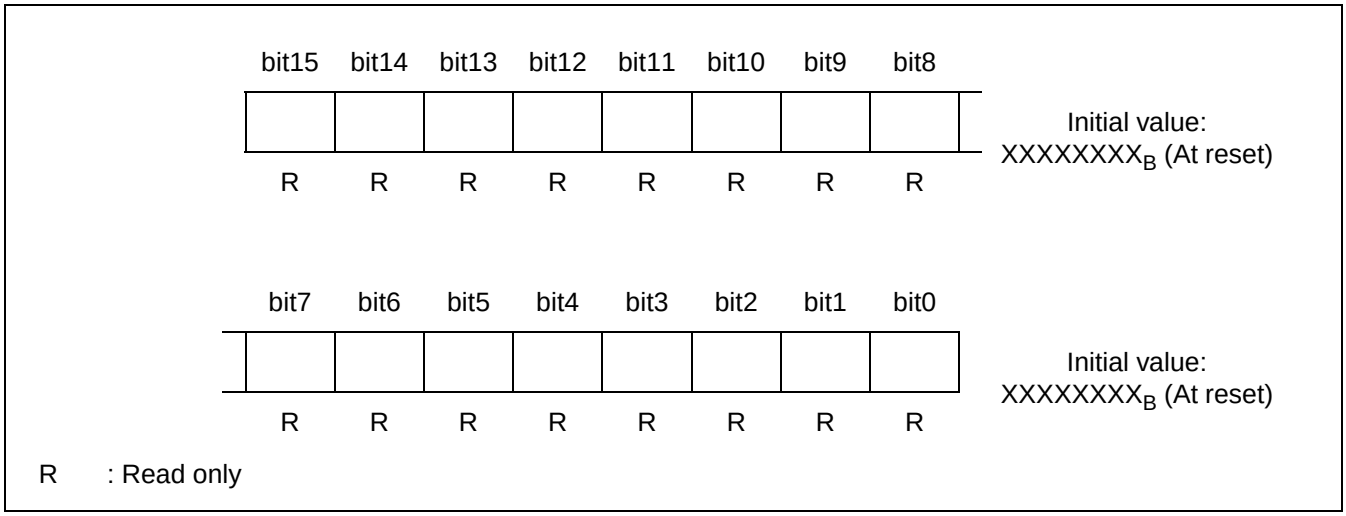
24.8.4.2 Data Buffer Register (BTxDTBF)

The data buffer register (BTxDTBF) allows the measured value or count value of the PWC timer to be read from. In 32-bit mode, the register holds the value of the lower 16 bits for the even-numbered channel or the value of the upper 16 bits for the odd-numbered channel.
To read this register, be sure to use a 16-bit data transfer instruction.

■ Bit Configuration of the Data Buffer Register (BTxDTBF)

Figure 24.8-37 shows the bit configuration of the data buffer register (BTxDTBF).

Figure 24.8-37 Bit Configuration of the Data Buffer Register (BTxDTBF)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8
Initial value:
XXXXXXXX$_B$ (At reset)
R R R R R R R R
bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
Initial value:
XXXXXXXX$_B$ (At reset)
R R R R R R R R
R : Read only

- The BTxDTBF register can only be read in both of the continuous and one-shot measurement modes. An attempt to write to the register makes no change to the register value.
- In continuous measurement mode (BTxTMCR: bit3 MDSE = 1), the BTxDTBF register serves as a buffer register holding the preceding measurement result.
- In one-shot measurement mode (BTxTMCR: bit3 MDSE = 0), the BTxDTBF register directly accesses the up counter. Even during counting, the count value can be read from this register. When the measurement is completed, the register preserved the measurement result as it is.
- Access the BTxDTBF register using 16-bit data.

24.8.4.3 PWC Operation

The PWC timer has a pulse width measurement feature, capable of selecting the count clock from among five types and measuring the time between arbitrary events of the input pulse and their cycle. The following outlines the basic functions and operations of the pulse width measurement feature.

■ Pulse Width Measurement Feature

When started, the timer clears the counter to "0000_H" but does not perform counting until the pre-set measurement start edge is input. Upon detection of the measurement start edge, the timer increments the counter from "0001_H". Upon detection of the measurement end edge, the timer stops the counter. The timer saves the count value between the two events as the pulse width to the register.

An interrupt request can be generated upon completion of measurement or when an overflow occurs.

After measurement, the timer acts as follows depending on the measurement mode:

- In one-shot measurement mode:The timer stops operation.
- In continuous measurement mode:The timer transfers the counter value to the buffer register and stops counting until the measurement start edge is input again.

Figure 24.8-38 Pulse Width Measurement Operation (One-shot Measurement Mode/"H" Width Measurement)

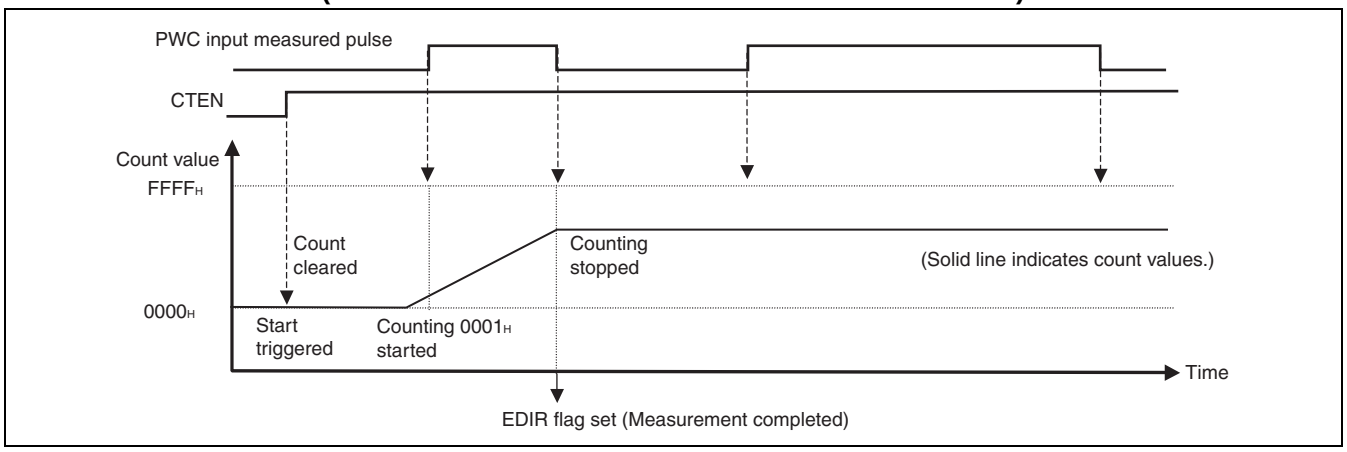
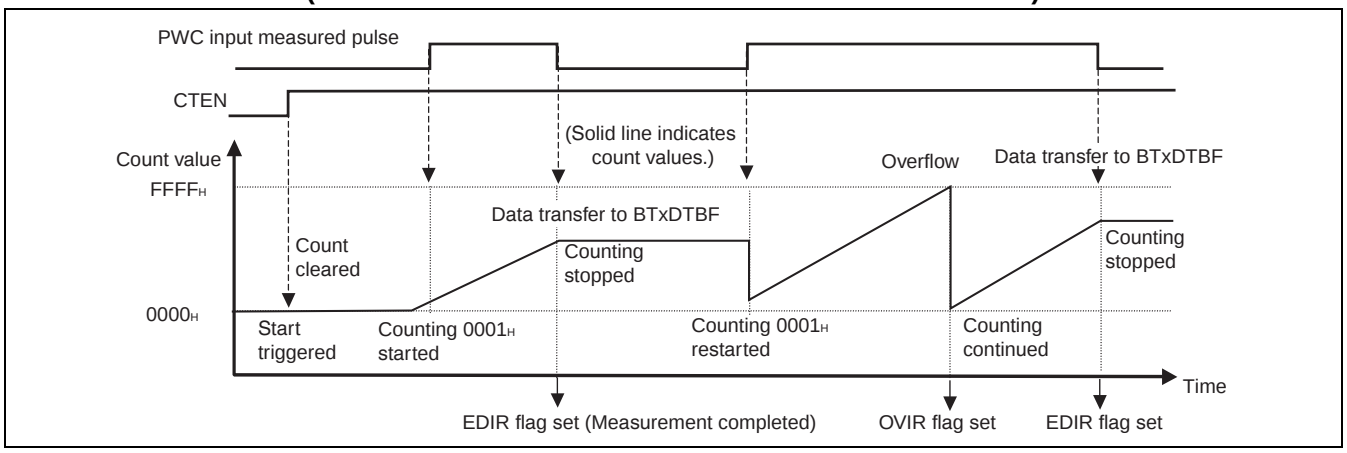


Figure 24.8-39 Pulse Width Measurement Operation (Continuous Measurement Mode/"H" Width Measurement)



■ Selecting the Count Clock

The count clock for the counter can be selected from among five types, depending on the settings of the CKS2 (bit6), CKS1 (bit5), and CKS0 (bit4) in the BTxTMCR registers.

The following count clocks can be selected:

BTxTMCR Register CKS2, CKS1, CKS0 bits	Internal count clock selected
000_B	Peripheral clock (PCLK) [Initial value]
001_B	Peripheral clock (PCLK) divided by 4
010_B	Peripheral clock (PCLK) divided by 16
011_B	Peripheral clock (PCLK) divided by 128
100_B	Peripheral clock (PCLK) divided by 256
101_B	Setting not allowed
110_B	
111_B	

The initial value immediately after a reset selects the peripheral clock (PCLK).

Note: Be sure to select the count clock before starting the counter.

■ Selecting the Operation Mode

Operation and measurement modes are selected depending on their settings in the BTxTMCR register.

Operation mode setting . . . . . BTxTMCR bit10 to bit8: EGS2, EGS1. EGS0
(Selecting the measurement edge)

Measurement mode setting . . . BTxTMCR bit2: MDSE
(Selecting one-shot/continuous measurement)

Listed below are the selectable operation modes and their respective bit settings.

Operation mode		MDSE	EGS2	EGS1	EGS0
↑ to ↓ "H" pulse width measurement	Continuous measurement: Buffer enabled	0	0	0	0
	One-shot measurement: Buffer disabled	1	0	0	0
↑ to ↑ measurement of period between rising edges	Continuous measurement: Buffer enabled	0	0	0	1
	One-shot measurement: Buffer disabled	1	0	0	1
↓ to ↓ measurement of period between falling edges	Continuous measurement: Buffer enabled	0	0	1	0
	One-shot measurement: Buffer disabled	1	0	1	0
↑ or ↓ to ↓ or ↑ measurement between all edges	Continuous measurement: Buffer enabled	0	1	1	1
	One-shot measurement: Buffer disabled	1	1	1	1
↓ to ↑ "L" pulse width measurement	Continuous measurement: Buffer enabled	0	1	0	0
	One-shot measurement: Buffer disabled	1	1	0	0
Setting not allowed		0	1	0	1
		1	1	0	1
		0	1	1	0
		1	1	1	0
		0	1	1	1
		1	1	1	1

The initial value immediately after a reset selects "H" pulse width/one-shot measurement mode.

Be sure to select the operation mode before starting the counter.

■ Starting and Stopping Pulse Width Measurement

Each type of measurement can be started, restarted, and aborted by the CTEN bit (bit1) in the BTxTMCR register.

You can start/restart pulse width measurement by writing "1" to the CTEN bit. You can abort it by writing "0" to the CTEN bit.

CTEN	Function
1	Starts/restarts pulse width measurement
0	Aborts pulse width measurement

■ Operation after being Started

The timer operation after the pulse width measurement mode has been started does not start counting until the measurement start edge is input. Upon detection of the measurement start edge, the 16-bit up counter starts counting from "0001_H".

■ Restarting

Restarting the timer means starting the timer during operation again while it has already been started (by writing "1" again to the CTEN bit already containing "1"). When restarted, the timer behaves as follows:

- If restarted the timer waiting for the measurement start edge: No effect on its operation.
- If restarted during measurement:The timer clears the counter to "0000_H" and waits for the measurement start edge again. If the restart and measurement end edge detection occur at the same time, the measurement end flag (EDIR) is set. In continuous measurement mode, the measurement result is transferred to the BTxDTBF register.

■ Stopping

In one-shot measurement mode, the timer stops counting automatically when the counter causes an overflow or when measurement is completed, requiring no special attention. To stop the timer either in continuous measurement mode or before it stops automatically, you have to abort it.

■ Clearing the Counters and Their Initial Values

The 16-bit up counter is cleared to "0000_H" when:

- a reset occurs
- "1" is written to the CTEN bit (bit1) in the BTxTMCR register (including the case of restarting).

The 16-bit up counter is initialized to "0001_H" when measurement start edge is detected.

■ Details of Pulse Width Measurement Operation

● One-shot measurement and continuous measurement

There are two modes of pulse width measurement: one is to perform measurement only once and the other is to perform measurement continuously. Each mode is selected by using the MDSE bit in the BTxTMCR register (see "■ Selecting the Operation Mode" in "24.8.4.3 PWC Operation"). The two modes have the following differences:

One-shot measurement mode:

When the measurement end edge is input once, the counter stops counting and the measurement end flag (EDIR) in the BTxSTC register is set, finishing the current measurement session. If the counter is restarted at the same time, however, it waits for the measurement start edge.

Continuous measurement mode:

When the measurement end edge is input, the counter stops counting, the measurement end flag (EDIR) in the BTxSTC register is set, and the counter remains idle until the measurement start edge is input again. Next time the measurement start edge is input, the counter is initialized to "0001_H" to start measurement. Upon completion of measurement, the measurement result in the counter is transferred to the BTxDTBF register.

Be sure to select or change the measurement mode with the counter stopped.

● Measurement result data

The one-shot measurement and continuous measurement modes are different in the handling of measurement results and counter values and the BTxDTBF function. The differences in measurement results between the two modes are as follows:

One-shot measurement mode:

When the BTxDTBF register is read during operation, the count value being measured can be obtained.

When the BTxDTBF register is read after measurement is completed, measurement result data is obtained.

Continuous measurement mode:

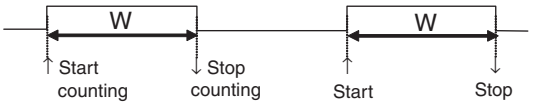
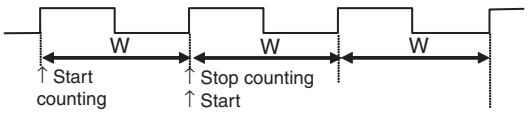
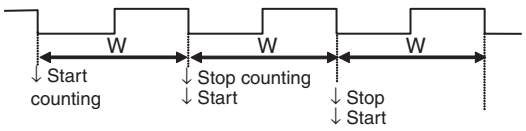
When measurement is completed, the measurement result in the counter is transferred to the BTxDTBF register.

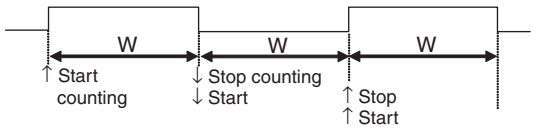
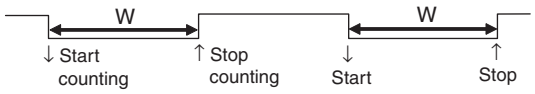
When the BTxDTBF register is read, the last measurement result is obtained. During measurement operation, the BTxDTBF register holds the result of preceding measurement. The count value being measured cannot be read.

If the current measurement is completed before the preceding measurement result is read in continuous measurement mode, the preceding measurement result is overwritten by the new measurement result. In this case, the error flag (ERR) in the BTxSTC register is set. The error flag (ERR) is cleared automatically when the BTxDTBF register is read.

■ Measurement Mode and Counting

Measurement mode can be selected from among five types, depending on what part of the input pulse is measured. The following table summarizes each measurement mode and its target.

Measurement mode	EGS2, EGS1, EGS0	Measurement target (W: Pulse width to be measured)
"H" pulse width measurement	000_B	 Measure the width of "H" period. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of falling edge
Measurement of period between rising edges	001_B	 Measure the period between rising edges. Start counting (measurement) : upon detection of rising edge Stop counting (measurement) : upon detection of rising edge
Measurement of period between falling edges	010_B	 Measure the period between falling edges. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of falling edge

Measurement mode	EGS2, EGS1, EGS0	Measurement target (W: Pulse width to be measured)
Measurement of pulse widths between all edges	011_B	 Measure the width between continuously input edges. Start counting (measurement) : upon detection of edge Stop counting (measurement) : upon detection of edge
Measurement of "L" pulse width	100_B	 Measure the width of the "L" period. Start counting (measurement) : upon detection of falling edge Stop counting (measurement) : upon detection of rising edge

In any measurement mode, the counter started for measurement is cleared to "0000_H" and remains idle without counting until the measurement start edge is input. When the measurement start edge is input, the counter is incremented every count clock until the measurement end edge is input.

When measurement of pulse widths between all edges or period measurement is performed in continuous measurement mode, the end edge becomes the next measurement start edge.

● Pulse width/period calculation method

The following equation can be used to calculate the measured pulse width/period from measurement result data obtained from the BTxDTBF register after measurement is completed:

$T_W = n \times t$ [ms]

T_W : Measured pulse width/period [ms]
n : Measurement result data in BTxDTBF
t : Count clock cycle [ms]

● Generating interrupt requests

Interrupt requests can be generated in two ways.

- Interrupt request in response to counter overflow

  When the counter is incremented to cause an overflow during measurement, the overflow flag (OVIR) is set and generates an interrupt request if overflow interrupt requests have been enabled.

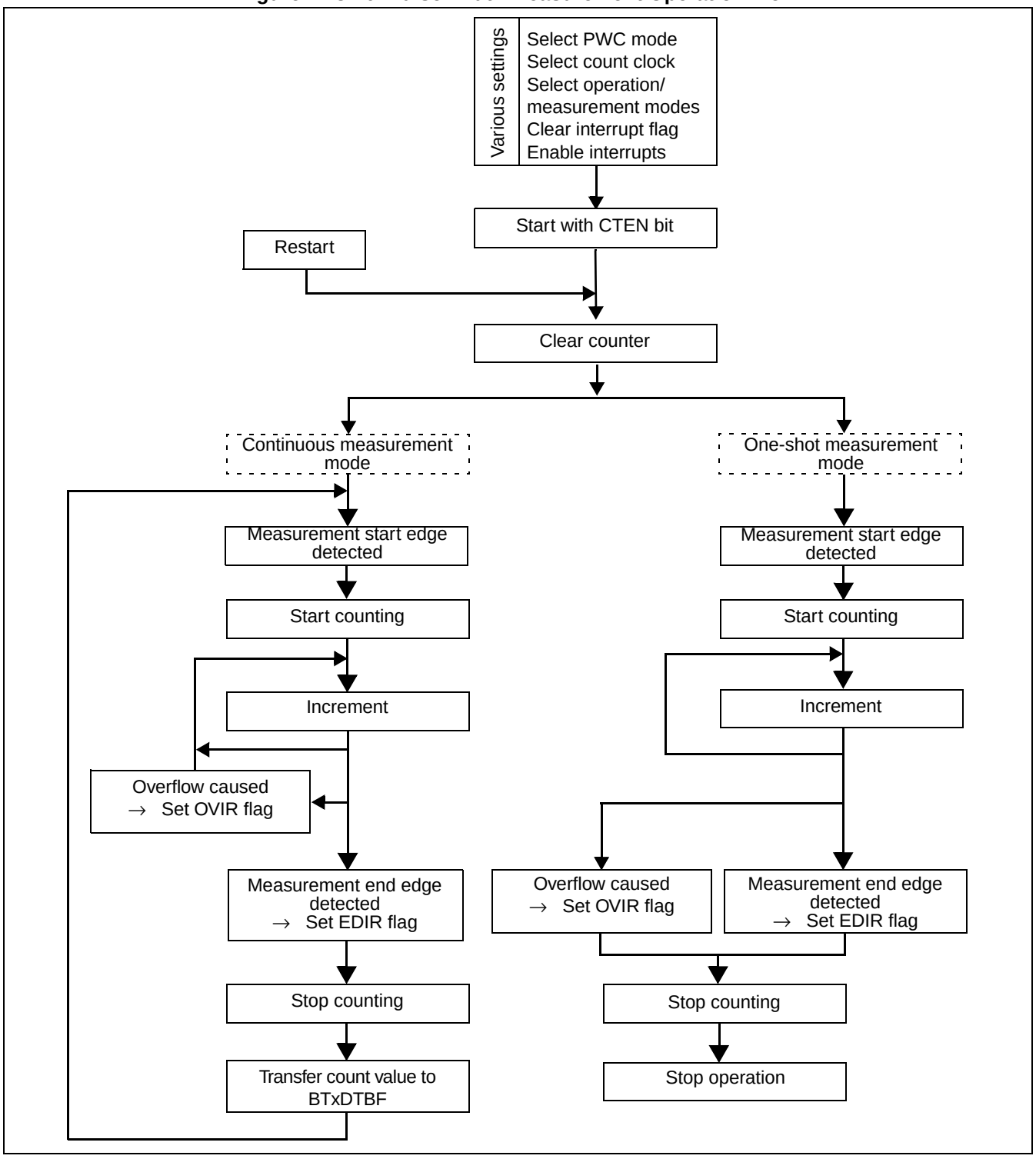
- Interrupt request upon completion of measurement

  When the measurement end edge is detected, the measurement end flag (EDIR) in the BTxSTC register is set and generates an interrupt request if measurement end interrupt requests have been enabled.

  The measurement end flag (EDIR) is cleared automatically when the measurement result is read from the BTxDTBF register.

■ Pulse Width Measurement Operation Flow

Figure 24.8-40 Pulse Width Measurement Operation Flow



CHAPTER 25 Up/Down Counter

This chapter explains the functions and operations of the up/down counter.

25.1 Overview
25.2 Configuration
25.3 Pin
25.4 Registers
25.5 Interrupt
25.6 An Explanation of Operations and Setting Procedure Examples

25.1 Overview

The up/down counter counts upward or downward depending on the setting.
By using only the lower byte of the 16-bit up/down counter, you can use it as an 8-bit up/down counter.
The counter can perform a count in a range of "00_H" to "FF_H" when used as an 8-bit up/down counter, and "0000_H" to "$FFFF_H$" when used as a 16-bit up/down counter.
This series microcontroller has 4 built-in channels for the 16-bit up/down counter. However, because only the lower byte can be used as an 8-bit up/down counter, you can use a total of 4 channels for both cases of using it as an 8-bit and a 16-bit counter.

■ Overview

- Counter mode: You can select the use of the counter either as an 8-bit up/down counter (8-bit mode), or as a 16-bit up/down counter (16-bit mode).
- Operation mode: One of the following three modes (4 types) can be selected.
  - Timer mode

    The counter counts downward by synchronizing with the count clock.

    The internal clock (peripheral clock) which is generated by dividing the peripheral clock (PCLK) by 2 or 8 by the prescaler is used as a count clock.
  - Up/Down count mode

    The counter counts upward/counts downward signals that are input from the 2 external signal input pins. You can select which edge to count from among the rising edge, falling edge, or both edges.
  - Phase difference count mode

    The counter counts upward/counts downward the phase difference of the signals that are input from the 2 external signal input pins.

    Phase difference count mode is appropriate for counting for the encoder of the motor and the like. Rotation angle and rotation number can be easily counted with high accuracy by inputting A-phase, B-phase, and Z-phase outputs respectively from the encoder.

    There are two phase difference count modes: one multiplied by 2-mode and one multiplied by 4-mode. The counting method for each of these modes differs from the other.

  Table 25.1-1 outlines the operation mode of the up/down counter.

Table 25.1-1 Operation mode of the up/down counter

Operation Mode	Count Timing	Count Direction
Timer mode	Internal clock (peripheral clock)	Count downward
Up/Down count mode	External clock	Count upward/Count downward
Phase difference count mode (Multiplied by 2/Multiplied by 4)	Phases of the input signals from the external signal input pins	Count upward/Count downward

- Reload compare function: One of the following three types can be selected.
  - Compare function

    When counting reaches the previously set value, the value of the counter is cleared to continue counting.
  - Reload function

    If an underflow occurs, the reload value is loaded to continue counting.
  - Reload compare function

    Compare function and reload function can be combined for use.
- Count direction: The last count direction (count upward/count downward) can be verified.
- Interrupt request: Can be generated in the following cases:
  - The count direction is inverted
  - The value of the counter matches the previously set value.
  - An overflow occurs
  - An underflow (reload) occurs

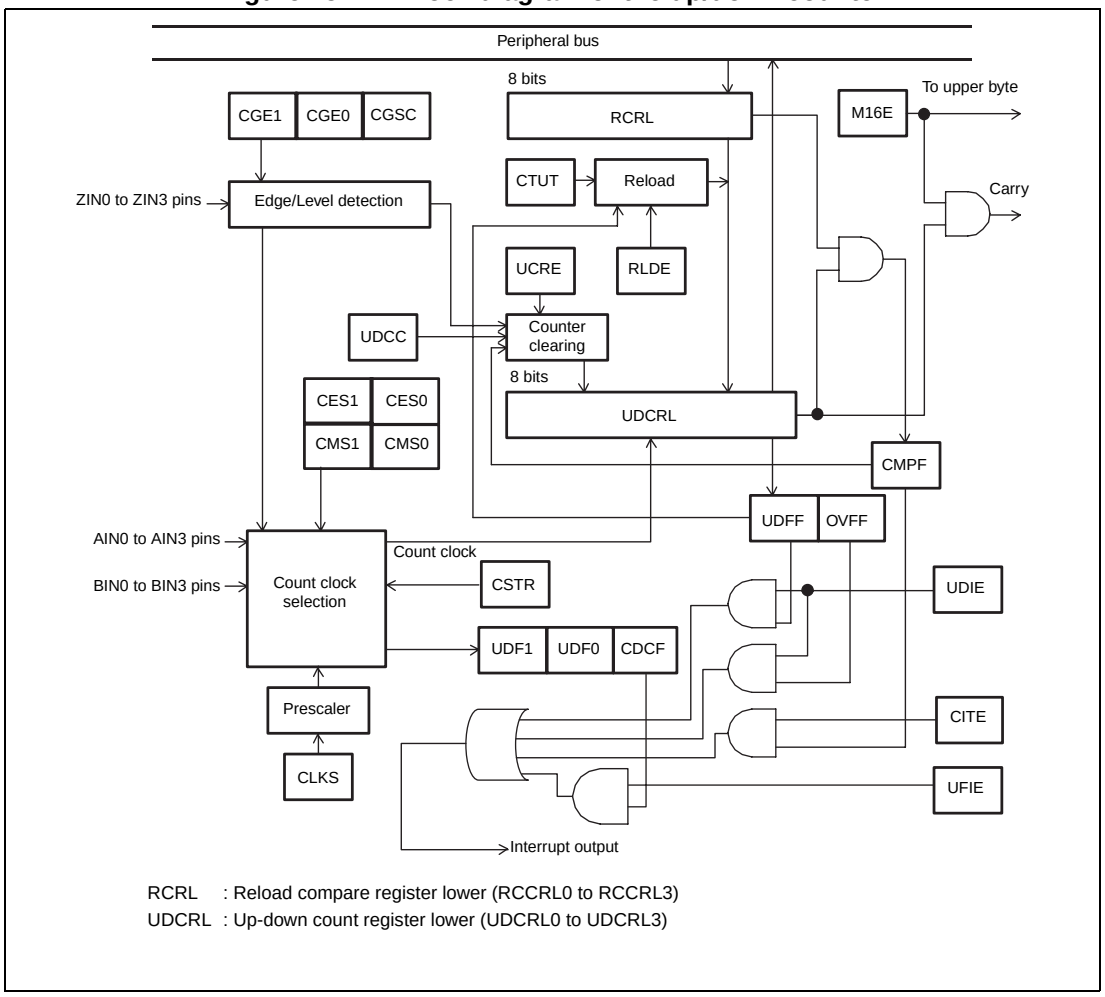
25.2 Configuration

This section shows the configuration of the up/down counter.

■ Block diagram of the up/down counter

Figure 25.2-1 is a block diagram of the up/down counter, taking ch.0 as an example.

Figure 25.2-1 Block diagram of the up/down counter



- Reload compare register (RCR0 to RCR3)

  This register sets the reload value and the compare value of the up/down counter.

  It is divided into the upper 8 bits and lower 8 bits as follows:

  The lower bits are used when the counter is used in 8-bit mode.

  - Reload compare register upper (RCRH0 to RCRH3)
  - Reload compare register lower (RCRL0 to RCRL3)

- Up-down count register (UDCR0 to UDCR3)

  This register operates as a counter of the up/down counter.

  It is divided into the upper 8 bits and lower 8 bits as follows:

  The lower bits are used when the counter is used in 8-bit mode.

  - Up-down count register upper(UDCRH0 to UDCRH3)
  - Up-down count register lower (UDCRL0 to UDCRL3)
- Counter control register (CCR0 to CCR3)

  This register controls the up/down counter.
- Counter status register (CSR0 to CSR3)

  This register verifies the state of the up/down counter and controls interrupt requests.
- Count clock selection circuit

  This circuit is used to select the count clock for the up/down counter.
- Prescaler

  This is used to select the division rate of the peripheral clock (PCLK) when the up/down counter is used in timer mode.

■ Clock

Table 25.2-1 shows the clock used by the up/down counter.

Table 25.2-1 Clock used by the up/down counter

Clock Name	Description	Remarks
Operation clock	Peripheral clock (PCLK)	-
Count clock	Internal clock (peripheral clock)	Generated through division of the peripheral clock (PCLK).
	Counts inputs from the external pins	Inputs from AIN0 to AIN3 pins and BIN0 to BIN3 pins

25.3 Pin

This section explains the pins of the up/down counter.

■ Overview

The up/down counter has the following three types of pins.

- AIN0 to AIN3 pins

  These are the external signal input pins of the up/down counter In up/down count mode, signals are counted upward if an effective edge is detected in these pins. In phase difference count mode (multiplied by 2/multiplied by 4), the phase difference between these pins and BIN0 to BIN3 pins is counted.

  These pins are multiplexed pins. To use them as AIN0 to AIN3 pins of the up/down counter, see "2.4 Setting Method for Pins".

- BIN0 to BIN3 pins

  These are the external signal input pins of the up/down counter In up/down count mode, signals are counted downward if an effective edge is detected in these pins. In phase difference count mode (multiplied by 2/multiplied by 4), the phase difference between these pins and AIN0 to AIN3 pins is counted.

  These pins are multiplexed pins. To use them as BIN0 to BIN3 pins of the up/down counter, see "2.4 Setting Method for Pins".

- ZIN0 to ZIN3 pins

  These are the external signal input pins of the up/down counter They are used to clear the counter or for gate input.

  These pins are multiplexed pins. To use them as ZIN0 to ZIN3 pins of the up/down counter, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 25.3-1 outlines the relationship between channels and pins.

Table 25.3-1 Relationship between Channels and Pins

Channel	External Signal Input Pin		
0	AIN0	BIN0	ZIN0
1	AIN1	BIN1	ZIN1
2	AIN2	BIN2	ZIN2
3	AIN3	BIN3	ZIN3

25.4 Registers

This section explains the configuration and functions of registers used by the up/down counter.

■ List of registers for the up/down counter

Table 25.4-1 lists the registers used by the up/down counter.

Table 25.4-1 Registers for the up/down counter

Channel	Abbreviated Register Name	Register Name	Reference
0	RCRL0	Reload compare register lower 0	25.4.1
	RCRH0	Reload compare register upper 0	25.4.1
	UDCRL0	Up-down count register lower 0	25.4.2
	UDCRH0	Up-down count register upper 0	25.4.2
	CCR0	Counter control register 0	25.4.3
	CSR0	Counter status register 0	25.4.4
1	RCRL1	Reload compare register lower 1	25.4.1
	RCRH1	Reload compare register upper 1	25.4.1
	UDCRL1	Up-down count register lower 1	25.4.2
	UDCRH1	Up-down count register upper 1	25.4.2
	CCR1	Counter control register 1	25.4.3
	CSR1	Counter status register 1	25.4.4
2	RCRL2	Reload compare register lower 2	25.4.1
	RCRH2	Reload compare register upper 2	25.4.1
	UDCRL2	Up-down count register lower 2	25.4.2
	UDCRH2	Up-down count register upper2	25.4.2
	CCR2	Counter control register 2	25.4.3
	CSR2	Counter status register 2	25.4.4
3	RCRL3	Reload compare register lower 3	25.4.1
	RCRH3	Reload compare register upper 3	25.4.1
	UDCRL3	Up-down count register lower 3	25.4.2
	UDCRH3	Up-down count register upper 3	25.4.2
	CCR3	Counter control register 3	25.4.3
	CSR3	Counter status register 3	25.4.4

25.4.1 Reload Compare Register (RCR0 to RCR3)

This register sets the reload value and the compare value of the up/down counter.
The reload value is a starting value to count downward with, and the compare value is a value to be compared with the counted value when counting upward (i.e., counting up is performed until the counted value reaches the compare value). The reload value and the compare value are the same.

This register is divided into upper byte and lower byte as follows:

- Reload compare register upper (RCRH0 to RCRH3)
- Reload compare register lower (RCRL0 to RCRL3)

In 16-bit mode, both the upper and lower byte values are used, while in 8-bit mode, the lower byte value is used.

By transferring the value that is written in this register to the up-down count register (UDCR0 to UDCR3), the up/down counter performs the count in a range from "0000_H" (for 8-bit mode, "00_H") to the value that has been set for this register.

Figure 25.4-1 shows the bit configuration of the reload compare register (RCR0 to RCR3).

Figure 25.4-1 Bit configuration of the reload compare register (RCR0 to RCR3)

Reload compare register upper (RCRH0 to RCRH3)

bit	15	14	13	12	11	10	9	8
	D15	D14	D13	D12	D11	D10	D9	D8
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

Reload compare register lower (RCRL0 to RCRL3)

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	W	W	W	W	W	W	W	W
Initial value	0	0	0	0	0	0	0	0

W: Write only

<Notes>

- By writing "1" to the CTUT bit of the counter control register (CCR0 to CCR3), the value that has been set for this register can be transferred to the up-down count register (UDCR0 to UDCR). However, note that the CTUT bit of the counter control register (CCR0 to CCR3) should be written while the up-down counter is stopped.
- If 16-bit mode is set in M16E bit (M16E = 1) of the counter control register (CCR0 to CCR3), this register must be written in half word.
- If 8-bit mode is set in M16E bit (M16E = 0) of the counter control register (CCR0 to CCR3), the reload compare register lower (RCRL0 to RCRL3) must be written in byte notation.

25.4.2 Up-Down Count Register (UDCR0 to UDCR3)

This register operates as a counter of the up/down counter. Also, the register can be read to verify the counter value.

This register is divided into upper byte and lower byte as follows:

- Up-down count register upper (UDCRH0 to UDCRH3)
- Up-down count register lower (UDCRL0 to UDCRL3)

In 8-bit mode, the upper byte value is invalid. Read the value of the up-down count register lower (UDCRL0 to UDCRL3).

Figure 25.4-2 shows the bit configuration of the up-down count register (UDCR0 to UDCR3).

Figure 25.4-2 Bit configuration of the up-down count register (UDCR0 to UDCR3)

Up-down count register upper (UDCRH0 to UDCRH3)

bit	15	14	13	12	11	10	9	8
	D15	D14	D13	D12	D11	D10	D9	D8
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

Up-down count register lower (UDCRL0 to UDCRL3)

bit	7	6	5	4	3	2	1	0
	D7	D6	D5	D4	D3	D2	D1	D0
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

<Notes>

- This register is read-only. To set a value to this register, transfer the value of the reload compare register (RCR0 to RCR3) to this register by using the following procedure.
  1. Write a value in the reload compare register (RCR0 to RCR3).
  2. Write "0" to the CSTR bit of the counter status register (CSR0 to CSR3).
  3. Write "1" to the CTUT bit of the counter control register (CCR0 to CCR3).
- If 16-bit mode is set in M16E bit (M16E = 1) of the counter control register (CCR0 to CCR3), this register must be read in half word.
- If 8-bit mode is set in the M16E bit (M16E = 0) of the counter control register (CCR0 to CCR3), the value of the up-down count register lower (UDCRL0 to UDCRL3) must be read.

25.4.3 Counter Control Register (CCR0 to CCR3)

This register controls operation of the up/down counter.

Figure 25.4-3 shows the bit configuration of the counter control register (CCR0 to CCR3).

Figure 25.4-3 Bit configuration of the counter control register (CCR0 to CCR3)

bit	15	14	13	12	11	10	9	8
	M16E	CDCF	CFIE	CLKS	CMS1	CMS0	CES1	CES0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	Reserved	CTUT	UCRE	RLDE	UDCC	CGSC	CGE1	CGE0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	1	0	0	0

R/W: Read/Write

R: Read only

[bit15]: M16E (16-bit mode selection bit)

This bit selects whether to use the up/down counter in 8-bit or 16-bit mode.

Written Value	Explanation
0	Uses the up/down counter in 8-bit mode (1 channel).
1	Uses the up/down counter in 16-bit mode (1 channel).

[bit14]: CDCF (Count direction change flag bit)

This bit indicates that the count direction is inverted from counting downward to counting upward or from counting upward to downward one or more times.

If the CFIE bit is set to "1" when this bit is "1", a count direction change interrupt request is generated.

CDCF	In Case of Reading	In Case of Writing
0	The count direction has not been inverted.	This bit is cleared to "0".
1	The count direction has been inverted one or more times.	Ignored

<Notes>

- If the counter reset occurs, the count direction is set to counting downward. Therefore, if counting upward is performed immediately after the reset, this bit changes to "1".
- If the count direction consecutively changes in a short period of time, the count direction may return to the original one with UDF1 and UDF0 bits of the counter status register (CSR0 to CSR3) unchanged.

[bit13]: CFIE (Count direction change interrupt enable bit)

This bit sets whether to generate the count direction change interrupt request if the count direction is inverted (CDCF = 1).

Written Value	Explanation
0	Disables generation of count direction change interrupt requests.
1	Enables generation of count direction change interrupt requests.

[bit12]: CLKS (Internal clock division selection bit)

This bit sets the division rate of the peripheral clock (PCLK) that is used as a count clock when timer mode is selected.

Written Value	Explanation
0	Peripheral clock (PCLK) divided by 2
1	Peripheral clock (PCLK) divided by 8

<Note>

This bit is enabled only when timer mode is set for the operation mode by setting the CMS1 and CMS0 bits (CMS1, CMS0 = 00). The setting of this bit is ignored if other operation modes are selected.

[bit11, bit10]: CMS1, CMS0 (Operation mode selection bit)

These bits select the operation mode of the up/down counter from among the following options.

- Timer mode

  The counter counts downward by synchronizing with the count clock.
- Up/Down count mode

  The counter counts upward/counts downward signals that are input from the 2 external signal input pins.
- Phase difference count mode

  The counter counts upward/counts downward the phase difference between the 2 external signal input pins. There are two phase difference count modes: one multiplied by 2-mode and one multiplied by 4-mode. The counting method for each of these modes differs from the other.

CMS1	CMS0	Operation Mode
0	0	Timer mode
0	1	Up/Down count mode
1	0	Phase difference count mode (multiplied by 2)
1	1	Phase difference count mode (multiplied by 4)

[bit9, bit8]: CES1, CES0 (Count clock edge selection bit)

These bits select the detection edge for the AIN0 to AIN3 pins and BIN0 to BIN3 pins.

When up/down count mode is selected, the count operation is performed every time if the edge that has been selected for this bit is detected.

CES1	CES0	Detection Edge
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both edges

<Note>

This bit is enabled only when up/down count mode is set for the operation mode by setting the CMS1 and CMS0 bits (CMS1, CMS0 = 01). The setting of this bit is ignored if other operation modes are selected.

[bit7]: Reserved bit

In Case of Writing	Always write "0" to this (these) bit (bits).
In Case of Reading	"0" is read.

[bit6]: CTUT (Counter write bit)

Transfers the values that have been set in the reload compare register (RCR0 to RCR3) to the up-down count register (UDCR0 to UDCR3).

CTUT	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Transfers the value.	

<Note>

The value of the reload compare register (RCR0 to RCR3) is transferred at the time when "1" is written to this bit. Therefore do not change this bit to "1" while the CSTR bit of the counter status register (CSR0 to CSR3) is "1" (the counter is active).

[bit5]: UCRE (Counter clear enable bit)

This bit enables/disables use of the compare function.

The compare function clears the counter value to "0000_H" to continue counting when the counter value matches the value set in the reload compare register (RCR0 to RCR3).

Written Value	Explanation
0	Disables use of the compare function.
1	Enables use of the compare function.

<Note>

This bit can control only the clear operation by using the compare function.

The following clear operations cannot be controlled by this bit.

- Clear operation by resetting this device
- Clear operation by effective edge inputs from the ZIN0 to ZIN3 pins (when CGSC bit = 0)
- Clear operation by writing "0" to the UDCC bit (clear by software)

[bit4]: RLDE (Reload enable bit)

This bit enables/disables use of the reload function.

The reload function reloads to the counter the value that has been set in the reload compare register (RCR0 to RCR3) when the counter underflows during count downward, and continues counting.

Written Value	Explanation
0	Disables use of the reload function.
1	Enables use of the reload function.

[bit3]: UDCC (Counter clear bit)

This bit clears the counter value to "0000_H".

UDCC	In Case of Writing	In Case of Reading
0	Clears the counter value.	"1" is read.
1	Ignored	

[bit2]: CGSC (Counter clear/Gate selection bit)

This bit selects the function for ZIN0 to ZIN3 pins from among the following options.

- Counter clear function

  The counter value is cleared to "0000_H" if the effective edge is input from the ZIN0 to ZIN3 pins.
- Gate function

  The counter operates only while the effective level is input from ZIN0 to ZIN3 pins.

Written Value	Explanation
0	Counter clear function
1	Gate function

<Note>

The ZIN0 to ZIN3 pins operate by combining settings of this bit and CGE1 and CGE0 bits. Be sure to also set CGE1 and CGE0 bits.

[bit1, bit0]: CGE1, CGE0 (Edge/Level selection bit)

These bits select the effective edge/effective level for the ZIN0 to ZIN3 pins. The meaning and function of these bits vary depending on the CGSC bit setting.

- When the counter clear function is selected in the CGSC bit (CGSC = 0)

  Selects the effective edge.

  The counter value is cleared to "0000_H" if the edge selected in this bit is detected in the ZIN0 to ZIN3 pins.
- When the gate function is selected in the CGSC bit (CGSC = 1)

  Selects the effective level.

  The counter operates only while the level selected in this bit is input from the ZIN0 to ZIN3 pins.

CGE1	CGE0	When the Counter Clear Function Is Selected (CGSC = 0)	When the Gate Function Is Selected (CGSC = 1)
0	0	Edge detection disabled	Level detection disabled (count disabled)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Setting prohibited	Setting prohibited

25.4.4 Counter Status Register (CSR0 to CSR3)

This register verifies the state of the up/down counter and controls interrupt requests.

Figure 25.4-4 shows the bit configuration of the counter status register (CSR0 to CSR3).

Figure 25.4-4 Bit configuration of the counter status register (CSR0 to CSR3)

bit	7	6	5	4	3	2	1	0
	CSTR	CITE	UDIE	CMPF	OVFF	UDFF	UDF1	UDF0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

[bit7]: CSTR (Count activation bit)

This bit activates/stops the up/down counter.

Written Value	Explanation
0	Stops count operation.
1	Activates the up/down counter.

[bit6]: CITE (Compare result match interrupt enable bit)

This bit sets whether to generate the comparison result match interrupt request if the counter value matches the value that has been set in the reload compare register (RCR0 to RCR3) (CMPF = 1).

Written Value	Explanation
0	Disables generation of comparison result match interrupt requests.
1	Enables generation of comparison result match interrupt requests.

[bit5]: UDIE (Overflow/Underflow interrupt enable bit)

This bit sets whether to generate the overflow/underflow interrupt request when the up/down counter overflows/underflows (OVFF/UDFF = 1).

Written Value	Explanation
0	Disables generation of overflow/underflow interrupt requests.
1	Enables generation of overflow/underflow interrupt requests.

[bit4]: CMPF (Compare result match detection flag bit)

This bit indicates that the counter value matches the value that has been set in the reload compare register (RCR0 to RCR3).

If the CITE bit is set to "1" when this bit is "1", the comparison result match interrupt request is generated.

CMPF	In Case of Reading	In Case of Writing
0	Values are not matched.	This bit is cleared to "0".
1	Values are matched.	Ignored

<Note>

The bit is changed to "1" in any of the following cases:

- The value matches during count upward.
- The value of the reload compare register (RCR0 to RCR3) is reloaded to the counter.
- Values are already matched when the up/down counter is activated.

[bit3]: OVFF (Overflow detection flag bit)

This bit indicates that the up/down counter overflows.

If the UDIE bit is set to "1" when this bit is "1", an overflow interrupt request is generated.

OVFF	In Case of Reading	In Case of Writing
0	No overflow occurred.	This bit is cleared to "0".
1	An overflow occurred.	Ignored

An overflow occurs when the counter value is "$FFFF_H$" and the counter attempts to count upward.

[bit2]: UDFF (Underflow detection flag bit)

This bit indicates that the up/down counter underflows.

If the UDIE bit is set to "1" when this bit is "1", an underflow interrupt request is generated.

UDFF	In Case of Reading	In Case of Writing
0	No underflow occurred.	This bit is cleared to "0".
1	An underflow occurred.	Ignored

An underflow occurs when the counter value is "0000_H" and the counter attempts to count downward.

[bit1, bit0]: UDF1, UDF0 (Up-down flag bit)

This bit indicates the last count direction.

This bit is updated each time the up/down counter performs a count operation.

UDF1	UDF0	Explanation
0	0	No input
0	1	Count downward
1	0	Count upward
1	1	Count upward/count downward concurrently

25.5 Interrupt

An interrupt request is generated in any of the following cases.

- The count direction is inverted (count direction change interrupt request)
- The counter value matches the value that has been set in the reload compare register (RCR0 to RCR3) (comparison result match interrupt request)
- An overflow occurs (overflow interrupt request)
- An underflow occurs (underflow interrupt request)

The generated interrupt request varies depending on the operation mode of the up/down counter. Table 25.5-1 outlines the relationship between the operation modes and interrupt requests.

Table 25.5-1 Relationship between the operation modes and interrupt requests

Interrupt Request	Timer Mode	Up/Down count mode	Phase difference count mode (Multiplied by 2/ multiplied by 4)
Count direction change interrupt request	x	O	O
Comparison result match interrupt request	O	O	O
Overflow interrupt request	x	O	O
Underflow interrupt request	O	O	O

Table 25.5-2 outlines the interrupts that can be used with the up/down counter.

Table 25.5-2 Interrupts of the up/down counter

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Clearing an Interrupt Request
Count direction change interrupt request	CDCF = 1 for CCR	CFIE = 1 for CCR	Write "0" to the CDCF bit in the CCR.
Comparison result match interrupt request	CMPF = 1 for CSR	CITE = 1 for CSR	Write "0" to the CMPF bit in the CSR.
Overflow interrupt request	OVFF = 1 for CSR	UDIE = 1 for CSR	Write "0" to the OVFF bit in the CSR.
Underflow interrupt request	UDFF = 1 for CSR	UDIE = 1 for CSR	Write "0" to the UDFF bit in the CSR.

CCR: Counter control register (CCR0 to CCR3)

CSR: Counter status register (CSR0 to CSR3)

<Notes>

- If an interrupt request is generated, the up/down counter is suspended until the interrupt request flag is cleared.
- The CMPF bit of the counter control register (CCR0 to CCR3) changes to "1" not only if the counted up value matches but also if the value has already been matched when the value of the reload compare register (RCR0 to RCR3) is reloaded or when the up/down counter is activated.
- For details of how to clear the counter and the reload timing, see "■ Clear event" and "■ Reload event" in "25.6 An Explanation of Operations and Setting Procedure Examples".
- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests.

  - Clears interrupt requests before enabling the generation of interrupt requests.
  - Clears interrupt requests simultaneously with interrupts enabled.
- For details of the interrupt vector number of the respective interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control register (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For interrupt level settings, see "CHAPTER 10 Interrupt Controller".

25.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operation of the up/down counter. Also, examples of procedures for setting the operating state are shown.

■ Overview

● Counter mode

The up/down counter can be used both as a 16-bit up/down counter and as an 8-bit up/down counter, depending on the setting.

This can be set in the M16E bit in the counter control register (CCR0 to CCR3).

- 8-bit mode (M16E = 0)

  Only the up-down count register lower (UDCRL0 to UDCRL3) is used. Write the reload value and compare value only in the reload compare register lower (RCRL0 to RCRL3) in byte notation.
- 16-bit mode (M16E = 1)

  Both the upper and lower bytes of the up-down count register (UDCR0 to UDCR3) are used. Write the reload value and compare value in the reload compare register (RCR0 to RCR3) in half word.

● Operation mode

One of the following three modes (4 types) can be selected for the operation mode of the up/down counter by using the CMS1 and CMS0 bit of the counter control register (CCR0 to CCR3).

- Timer mode (CMS1, CMS0 = 00)

  In this mode, counting downward is performed starting from the previously set value by synchronizing with the count clock.

  The count clock is generated by dividing the peripheral clock (PCLK) by 2 or 8 with the prescaler.
- Up/Down count mode (CMS1, CMS0 = 01)

  In this mode, signals that are input from the external signal input pins are counted upward or downward.
- Phase difference count mode (multiplied by 2) (CMS1, CMS0 = 10)/Phase difference count mode (multiplied by 4) (CMS1, CMS0 = 11)

  In this mode, phase difference between the signals that are input from the external signal input pins are counted upward or downward. By inputting A-phase of the encoder from the AIN0 to AIN3 pins, B-phase from the BIN0 to BIN3 pins, and Z-phase from the ZIN0 to ZIN3 pins, rotation angle and rotation number can be counted and rotation direction can be detected with high accuracy, making it appropriate for counting for the encoder of motors and the like.

■ Functions that can be used

● Reload/Compare function

8/16-bit up/down counter can enable and disable the reload function and compare function by using the RLDE bit and UCRE bit in the counter control register (CCR0 to CCR3).

- Reload function

  This function reloads the value that has been set in the reload compare register (RCR0 to RCR3) if an underflow occurs during count downward, and performs count downward again. For details of this operation, see "■ Count operation" in "25.6.1 Operation in Timer Mode".

- Compare function

  In this function, if an attempt is made to further count upward while the value of the up/down counter matches the value that has been set in the reload compare register (RCR0 to RCR3) (comparison result match), the value of the up/down counter is cleared to "0000_H" to start counting upward again. For details of this operation, see "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

  This function cannot be used in timer mode.

- Reload compare function

  this is a function used by combining the reload function and compare function. In this function, counting of any range is possible because counting upward/downward is performed between the values of "0000_H" and the value set in the reload compare register (RCR0 to RCR3). See "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

  This function cannot be used in timer mode.

Table 25.6-1shows how to set the reload function/compare function.

Table 25.6-1 Setting the reload/compare function

RLDE bit	UCRE bit	Explanation
0	0	Disable reload function/compare function
0	1	Disable reload function Enable compare function
1	0	Enable reload function Disable compare function
1	1	Enable reload compare function

● Function of the ZIN0 to ZIN3 pins

One of the following functions can be selected for the ZIN0 to ZIN3 pins using the CGSC bit of the counter control register (CCR0 to CCR3).

- Count clear function (CGSC = 0)

  The counter value is cleared to "0000_H" if an effective edge is input from the ZIN0 to ZIN3 pins during count operation.

- Gate function (CGSC = 1)

  The counter operates only when the effective level is being input from the ZIN0 to ZIN3 pins.

When the counter clear function is selected, select the effective edge. When the gate function is selected, select the effective level. Make these selections by using the CGE1 and CGE0 bits of the counter control register (CCR0 to CCR3).

CGE1	CGE0	When the Counter Clear Function Is Selected (CGSC = 0)	When the Gate Function Is Selected (CGSC = 1)
0	0	Edge detection disabled	Level detection disabled (count disabled)
0	1	Falling edge	"L" level
1	0	Rising edge	"H" level
1	1	Setting prohibited	Setting prohibited

■ Clear event

The counter value is cleared to "0000_H" in one of the following cases:

- This device is reset.
- The effective edge is input from the ZIN0 to ZIN3 pins.

  (When the counter clear function for the ZIN0 to ZIN3 pins is set in the CGSC bit (CGSC =0) of the counter control register (CCR0 to CCR3).
- Software clear

  "0" is written to the UDCC bit of the counter control register (CCR0 to CCR3).
- Clear with the compare function

  The counter value matches the value set in the reload compare register (RCR0 to RCR3) and the counter further attempts to count upward.

  (The count value is not cleared if counting downward is performed or the counter is stopped.)
- Clear with an overflow generation

  Timing of count upward/count downward after the counter value reaches "$FFFF_H$" (in 8-bit mode, "FF_H")

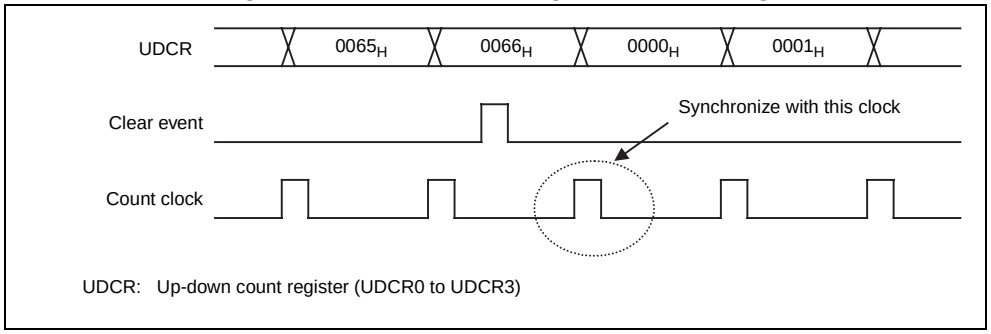
The timing of clearing the counter value to "0000_H" depends on the operation state of the up/down counter as follows.

- When a clear event occurs during count operation

  The value is cleared by synchronizing with the count clock.

  Figure 25.6-1 shows the timing for the clear event to occur.

Figure 25.6-1 Clear event generation timing

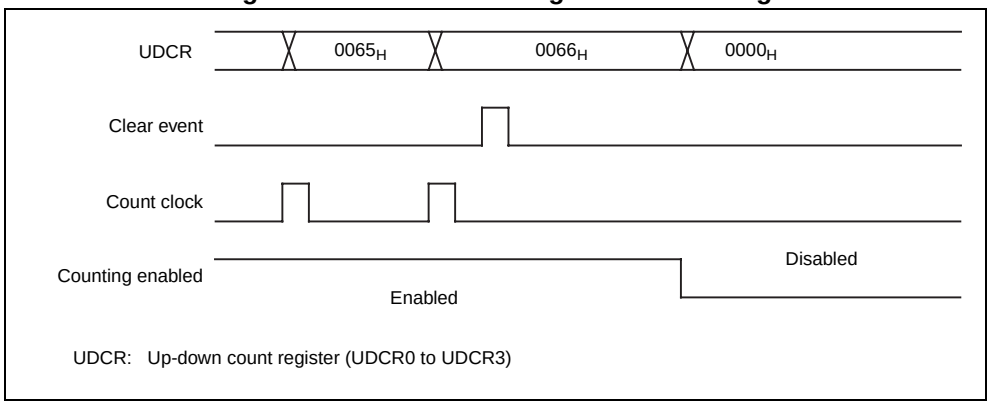


- When a clear event occurs during count operation and the count operation is stopped before the next count clock is input (CSTR bit = 0, in the counter status register (CSR0 to CSR3))

  The value is cleared at the point where the up/down counter stops.

  Figure 25.6-2 shows the clear event generation timing.

Figure 25.6-2 Clear event generation timing



■ Reload event

The value of the up/down counter is reloaded in any of the following cases.

- "1" is written to the CTUT bit of the counter control register (CCR0 to CCR3)
- The value is reloaded by the reload function

The timing at which the value of the up/down counter is reloaded is listed below, which depends on the operation state of the up/down counter.

- When the reload event occurs during count operation

  The value is reloaded by synchronizing with the count clock.
- When the reload event occurs while counting is stopped

  The value is reloaded at the time when the reload event occurs.

<Notes>

- Do not write "1" to the CTUT bit of the counter control register (CCR0 to CCR3) during count operation.
- If the reload event and clear event occur at the same time, the clear event has priority.

25.6.1 Operation in Timer Mode

This section explains operations in timer mode.

■ Overview

In this mode, counting downward is performed starting from the value that has been set in the reload compare register (RCR0 to RCR3). The peripheral clock (PCLK) is used as a count clock by dividing it with the prescaler.

You can also use the reload function, which reloads the value of the reload compare register (RCR0 to RCR3) when the counter underflows to restart counting downward.

■ Count operation

● Normal operation

1. Set the reload value/compare value in the reload compare register (RCR0 to RCR3).
2. Write "1" to the CTUT bit of the counter control register (CCR0 to CCR3).

   The set value is transferred to the up-down count register (UDCR0 to UDCR3).
3. Enable operation of the up/down counter by setting the CSTR bit (CSTR = 1) of the counter status register (CSR0 to CSR3).

   Counting downward starts from the value that has been set in the reload compare register (RCR0 to RCR3).

If the counter underflows, the UDFF bit of the counter status register (CSR0 to CSR3) changes to "1". At this point, if the UDIE bit of the counter status register is set to "1", the underflow interrupt request is generated.

If the gate function is set in the ZIN0 to ZIN3 pins by use of the CGSC bit (CGSC = 1) of a counter control register (CCR0 to CCR3), the counting is performed only when the effective level, which was set in the CGE1 and CGE0 bits, is input from the ZIN0 to ZIN3 pins.

For details of effective level settings, see "25.4.3 Counter Control Register (CCR0 to CCR3)".

<Note>

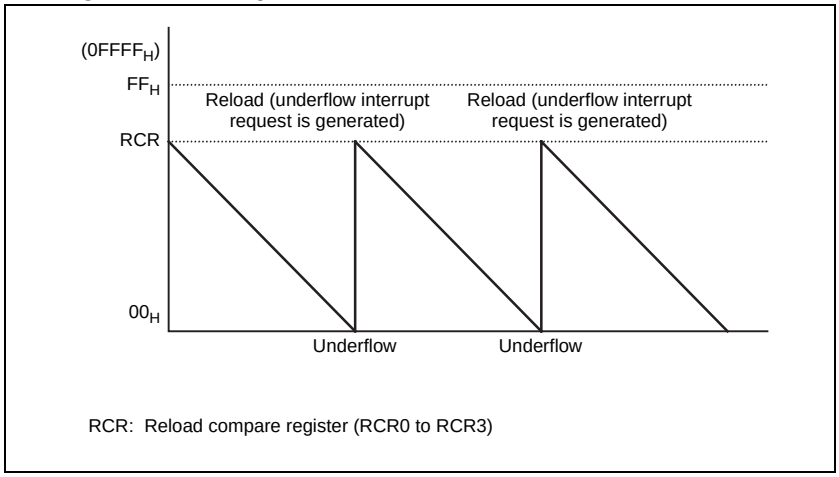
The minimum pulse width required for the ZIN0 to ZIN3 pins is 2T (T: period of the peripheral clock (PCLK)).

● Operation when the reload function is used

If the counter underflows while counting downward, the UDFF bit of the counter status register (CSR0 to CSR3) changes to "1". The value of the reload compare register (RCR0 to RCR3) is reloaded at the next timing of underflow occurrence and counting down is restarted. At this point, if the UDIE bit of the counter status register (CSR0 to CSR3) is set to "1", the underflow interrupt request is generated.

Figure 25.6-3 shows the operations when the reload function is used.

Figure 25.6-3 Operation when the reload function is used



<Note>

The value of the reload compare register (RCR0 to RCR3) is a reload value as well as a compare value. Therefore, if the value of the reload compare register (RCR0 to RCR3) is reloaded, the CMPF bit of the counter status register (CSR0 to CSR3) also changes to "1".

25.6.2 Operations in Up/Down Count Mode

This section explains operations in the up/down count mode.

■ Overview

In this mode, the external signals that are input from the AIN0 to AIN3 pins and BIN0 to BIN3 pins are counted upward/downward as a count clock.

Signals are counted upward if the external signals are input from the AIN0 to AIN3 pins and counted downward if input from the BIN0 to BIN3 pins.

One of the following edges can be selected for counting the external signals by setting the CES1 and CES0 bits of the counter control register (CCR0 to CCR3).

- Falling edge (CES1, CES0 = 01)
- Rising edge (CES1, CES0 = 10)
- Both edges (CES1, CES0 = 11)

In up/down count mode, the following three functions can be used.

- Reload function
- Compare function
- Reload compare function

■ Count operation

● Normal operation

While the counter is enabled, if the effective edge is input from the AIN0 to AIN3 pins, signals are counted upward, and if the effective edge is input from the BIN0 to BIN3 pins, signals are counted downward.

If the count direction is inverted such as from count up to count down or from count down to count up, the CDCF bit of the counter control register (CCR0 to CCR3) changes to "1". At this point, if "1" is set for the CFIE bit of the counter control register (CCR0 to CCR3), a count direction change interrupt request is generated.

If the gate function is set in the ZIN0 to ZIN3 pins by use of the CGSC bit (CGSC = 1) of a counter control register (CCR0 to CCR3), the counting is performed only when the effective level, which was set in the CGE1 and CGE0 bits, is input from the ZIN0 to ZIN3 pins.

For details of effective level settings, see "25.4.3 Counter Control Register (CCR0 to CCR3)".

<Note>

The minimum pulse width required for the AIN0 to AIN3 pins, BIN0 to BIN3 pins, and ZIN0 to ZIN3 pins is 2T (T: period of the peripheral clock (PCLK)).

● Operation when the reload function is used

The operation is the same as that in timer mode. See "■ Count operation" in "25.6.1 Operation in Timer Mode".

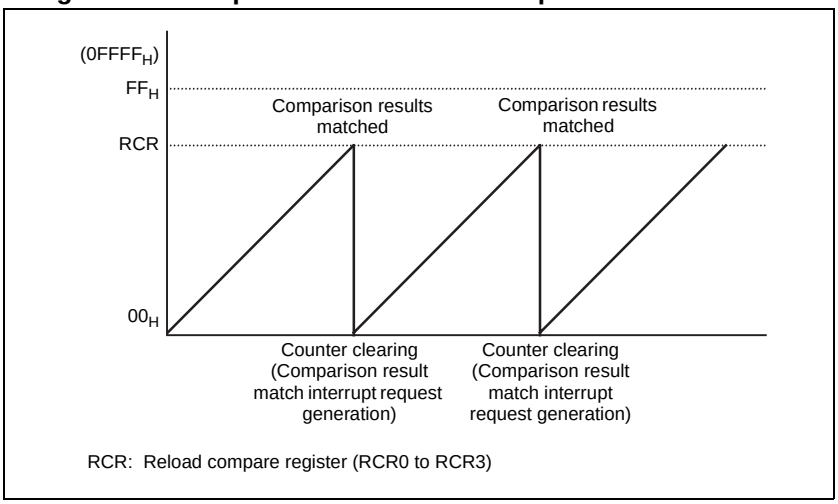
● Operations when the compare function is used

If the value of the up/down counter matches the value that has been set for the reload compare register (RCR0 to RCR3), the CMPF bit of the counter status register (CSR0 to CSR3) changes to "1". At this point, if CITE bit of the counter status register (CSR0 to CSR3) is set to "1", the comparison result match interrupt request is generated.

In this state, if an attempt to further count upward is performed, the value of the up/down counter is cleared to "0000_H" to restart counting upward.

Figure 25.6-4 shows the operations when the compare function is used.

Figure 25.6-4 Operations when the compare function is used



<Notes>

When the compare function is used, the value of the up/down counter is cleared to "0000_H" if the following conditions are met.

- The value of the up/down counter and the value that has been set in the reload compare register (RCR0 to RCR3) match (comparison result match)
- After that, another count has been performed.

However, the value of the up/down counter is not cleared in the following cases even if the comparison result matches.

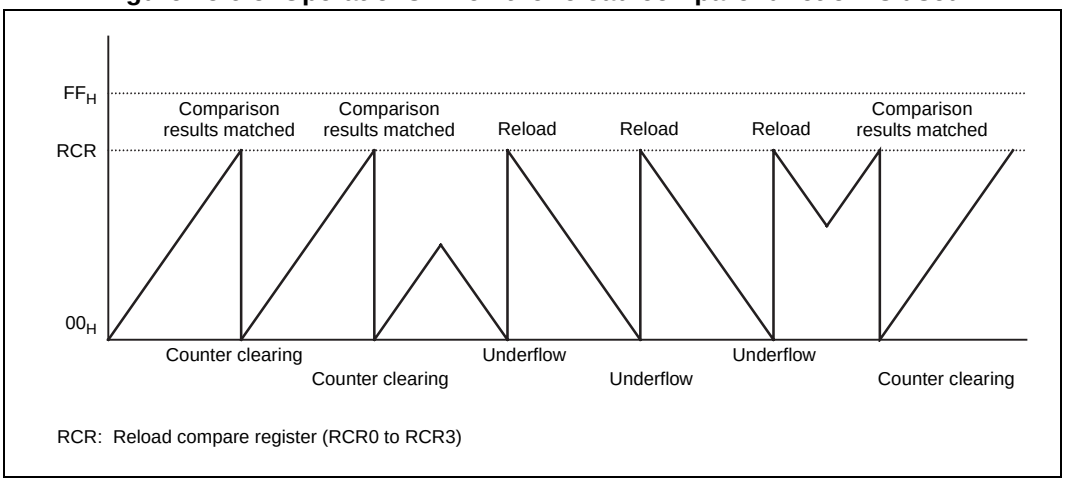
- The next operation is counting downward
- The up/down counter is stopped.

● Operations when the reload compare function is used

During count downward, the reload function is used, and during count upward, the compare function is used.

Figure 25.6-5 shows the operations when the reload compare function is used.

Figure 25.6-5 Operations when the reload compare function is used



■ Verifying the count direction

In this mode, both count upward and downward are performed. Therefore, the count direction can be verified with the UDF1 and UDF0 bits of the counter status register (CSR0 to CSR3). As these bits are rewritten each time counting is performed, they can be used to verify the current count direction. This is helpful if you want to know the rotation direction such as for controlling the motor.

Table 25.6-2 shows the count directions indicated with the UDF1 and UDF0 bits.

Table 25.6-2 Relationship between the UDF1 and UDF0 bits and count direction

UDF1	UDF0	Count Direction
0	0	No input
0	1	Count downward
1	0	Count upward
1	1	Count upward/count downward concurrently

If the count direction is inverted one or more times such as from count downward to count upward or from count upward to count downward, the CDCF bit of the counter control register (CCR0 to CCR3) changes to "1". At the time when this bit is changed, the count direction change interrupt request can also be generated. Thus, you can verify whether the count direction has been inverted by using the CDCF bit and generation of the count direction change interrupt request.

<Note>

If the count direction consecutively changes in a short period of time, the count direction may return to the original one with the UDF1 and UDF0 bits of the counter status register (CSR0 to CSR3) indicating the same direction as one indicated before the CDCF bit changed to "1".

25.6.3 Operations in Phase Difference Count Mode (Multiplied by 2)

This section explains operations in phase difference count mode (multiplied by 2).

■ Overview

In this mode, phase difference between the signals that are input from the 2 external signal input pins is counted. This mode is appropriate for counting phase difference between the A-phase and B-phase of the encoder output.

When the rising edge or falling edge is detected from the BIN0 to BIN3 pins, the input level of the AIN0 to AIN3 pins is verified to perform counting upward/downward the phase difference between the BIN0 to BIN3 pins and AIN0 to AIN3 pins. If the A-phase leads the B-phase, the counter counts upward; if the A-phase falls behind the B-phase, the counter counts downward.

Whether the counter counts upward or downward depends on the detection edge of the BIN0 to BIN3 pins and input level of the AIN0 to AIN3 pins.

Table 25.6-3 lists how to count.

Table 25.6-3 Counting method

BIN0 to BIN3 pins	AIN0 to AIN3 pins	Count Direction
Rising edge	"H" level	Count upward
	"L" level	Count downward
Falling edge	"H" level	Count downward
	"L" level	Count upward

In phase difference count mode (multiplied by 2), the following three functions can be used.

- Reload function
- Compare function
- Reload compare function

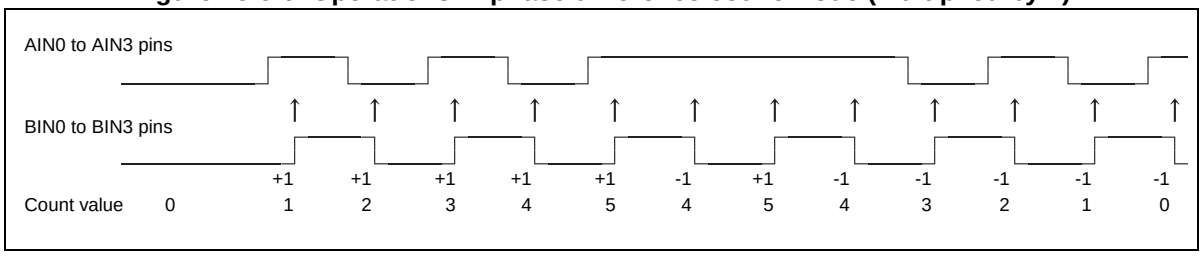
■ Count operation

● Normal operation

While the counter is enabled, if the rising edge/falling edge is input from the BIN0 to BIN3 pins, the input level of the AIN0 to AIN3 pins is detected to perform counting upward/downward.

Figure 25.6-6 shows the operations in phase difference count mode (multiplied by 2).

Figure 25.6-6 Operations in phase difference count mode (multiplied by 2)



If the gate function is set in the ZIN0 to ZIN3 pins by use of the CGSC bit (CGSC = 1) of a counter control register (CCR0 to CCR3), the counting is performed only when the effective level, which was set in the CGE1 and CGE0 bits, is input from the ZIN0 to ZIN3 pins.

For details of effective level settings, see "25.4.3 Counter Control Register (CCR0 to CCR3)".

<Note>

The minimum pulse width required for the AIN0 to AIN3 pins, BIN0 to BIN3 pins, and ZIN0 to ZIN3 pins is 2T (T: period of the peripheral clock (PCLK)).

● Operation when the reload function is used

The operation is the same as that in timer mode. See "■ Count operation" in "25.6.1 Operation in Timer Mode".

● Operations when the compare function is used

The operation is the same as that in up/down count mode. See "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

● Operations when the reload compare function is used

The operation is the same as that in up/down count mode. See "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

■ Verifying the count direction

The method is the same as that for the up/down count mode. See "■ Verifying the count direction" in "25.6.2 Operations in Up/Down Count Mode".

25.6.4 Operations in Phase Difference Count Mode (Multiplied by 4)

This section explains operations in phase difference count mode (multiplied by 4).

■ Overview

In this mode, the phase difference between the signals that are input from the 2 external signal input pins is counted. This mode is appropriate for counting phase difference between the A-phase and B-phase of the encoder output.

When the rising edge or falling edge is detected from the AIN0 to AIN3 pins or from the BIN0 to BIN3 pins, the input level of the other pins is verified to perform counting upward/downward the phase difference between the BIN0 to BIN3 pins and AIN0 to AIN3 pins.

Whether the counter counts upward or downward depends on the combination of the detected edge and input level.

Table 25.6-4 shows how to count.

Table 25.6-4 Counting method

Edge Detection Pin	Detected Edge	Level Verification Pin	Input Level	Count Direction
BIN0 to BIN3 pins	Rising edge	AIN0 to AIN3 pins	"H" level	Count upward
			"L" level	Count downward
	Falling edge		"H" level	Count downward
			"L" level	Count upward
AIN0 to AIN3 pins	Rising edge	BIN0 to BIN3 pins	"H" level	Count downward
			"L" level	Count upward
	Falling edge		"H" level	Count upward
			"L" level	Count downward

In phase difference count mode (multiplied by 4), the following three functions can be used.

- Reload function
- Compare function
- Reload compare function

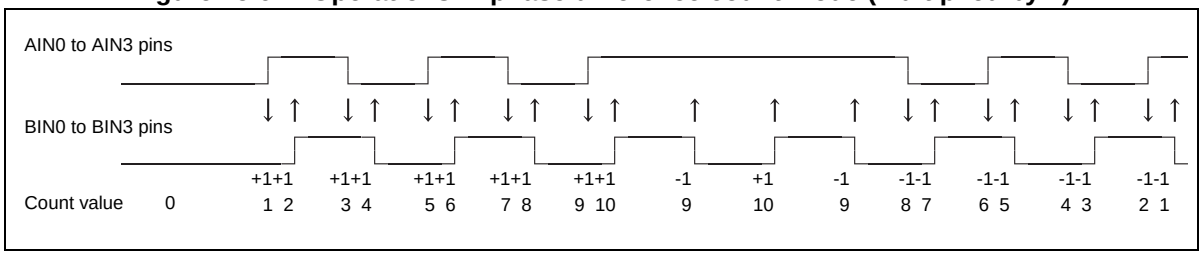
■ Count operation

● Normal operation

While the counter is enabled, if the rising edge/falling edge is input from the AIN0 to AIN3 pins or from the BIN0 to BIN3 pins, input level of the other pins is detected to perform counting upward/downward.

Figure 25.6-7 shows the operations in phase difference count mode (multiplied by 4).

Figure 25.6-7 Operations in phase difference count mode (multiplied by 4)



If the gate function is set in the ZIN0 to ZIN3 pins by use of the CGSC bit (CGSC = 1) of a counter control register (CCR0 to CCR3), the counting is performed only when the effective level, which was set in the CGE1 and CGE0 bits, is input from the ZIN0 to ZIN3 pins.

For details of effective level settings, see "25.4.3 Counter Control Register (CCR0 to CCR3)".

<Note>

The minimum pulse width required for the AIN0 to AIN3 pins, BIN0 to BIN3 pins, and ZIN0 to ZIN3 pins is 2T (T: period of the peripheral clock (PCLK)).

● Operation when the reload function is used

The operation is the same as that in timer mode. See "■ Count operation" in "25.6.1 Operation in Timer Mode".

● Operations when the compare function is used

The operation is the same as that in up/down count mode. See "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

● Operations when the reload compare function is used

The operation is the same as that in up/down count mode. See "■ Count operation" in "25.6.2 Operations in Up/Down Count Mode".

■ Verifying the count direction

The method is the same as that for the up/down count mode. See "■ Verifying the count direction" in "25.6.2 Operations in Up/Down Count Mode".

CHAPTER 26 10-Bit A/D Converter

This chapter explains the functions and operations of the 10-bit A/D converter.

26.1 Overview
26.2 Configuration
26.3 Pins
26.4 Registers
26.5 Interrupts
26.6 Explanation of Operations and Setting Procedure Examples

26.1 Overview

The 10-bit A/D converter is a device for converting analog signals to 10-bit digital signals.
This series microcontroller has 10-bit A/D converters, and 24 analog input channels can be assigned for conversion.

■ Overview

- Conversion time: 1.2 µs per channel, minimum (33 MHz peripheral clock (PCLK))
- Comparison/conversion method: RC-type successive comparison and conversion with sample-and-hold circuits
- Conversion mode: The modes that can be used are categorized into the following two types:
  - A/D scan conversion

    An optional conversion channel is selected from 24 channels and made subject to conversion.

    Two conversion modes are available: single conversion mode and repeat conversion mode. In single conversion mode, signals from the selected channel are converted only once. In repeat conversion mode, signals from the selected channel are converted repeatedly.
  - A/D priority conversion

    Once an activation trigger for high-priority A/D conversion is generated, that conversion is performed soon afterward by stopping A/D scan conversion in progress. There are two priority levels.
- Activation trigger: Activation triggers vary depending on the A/D conversion mode:
  - A/D scan conversion

    Conversion is activated by software or at detection of a rising edge of the TOUT signal of base timer ch.0.
  - A/D priority conversion (priority 1)

    Conversion is triggered by input of a falling edge from an external trigger input pin.
  - A/D priority conversion (priority 2)

    Conversion is activated by software or at detection of a rising edge of the TOUT signal of base timer ch.2.
- FIFO functionality: There 16 FIFO levels for A/D scan conversion and 4 FIFO levels for A/D priority conversion.
- Conversion result compare function: A/D conversion results can be compared.
- Independent control of channels: One of two kinds of sampling time can be set for each channel.
- Conversion results: A/D conversion results can specified to be stored left-justified (MSB side) or right-justified (LSB side).
- Interrupt request: Can be issued in the following cases:
  - Data has been stored in the predetermined number of stages in the FIFO used during A/D scan conversion.
  - Data has been stored in the predetermined number of stages in the FIFO used during A/D priority conversion.
  - A FIFO overrun occurred.
  - The comparison function is used to determine whether conversion results satisfy the interrupt request generation conditions.
- DMA transfer activation: Generation of an interrupt request can be used for DMA transfer of conversion results.

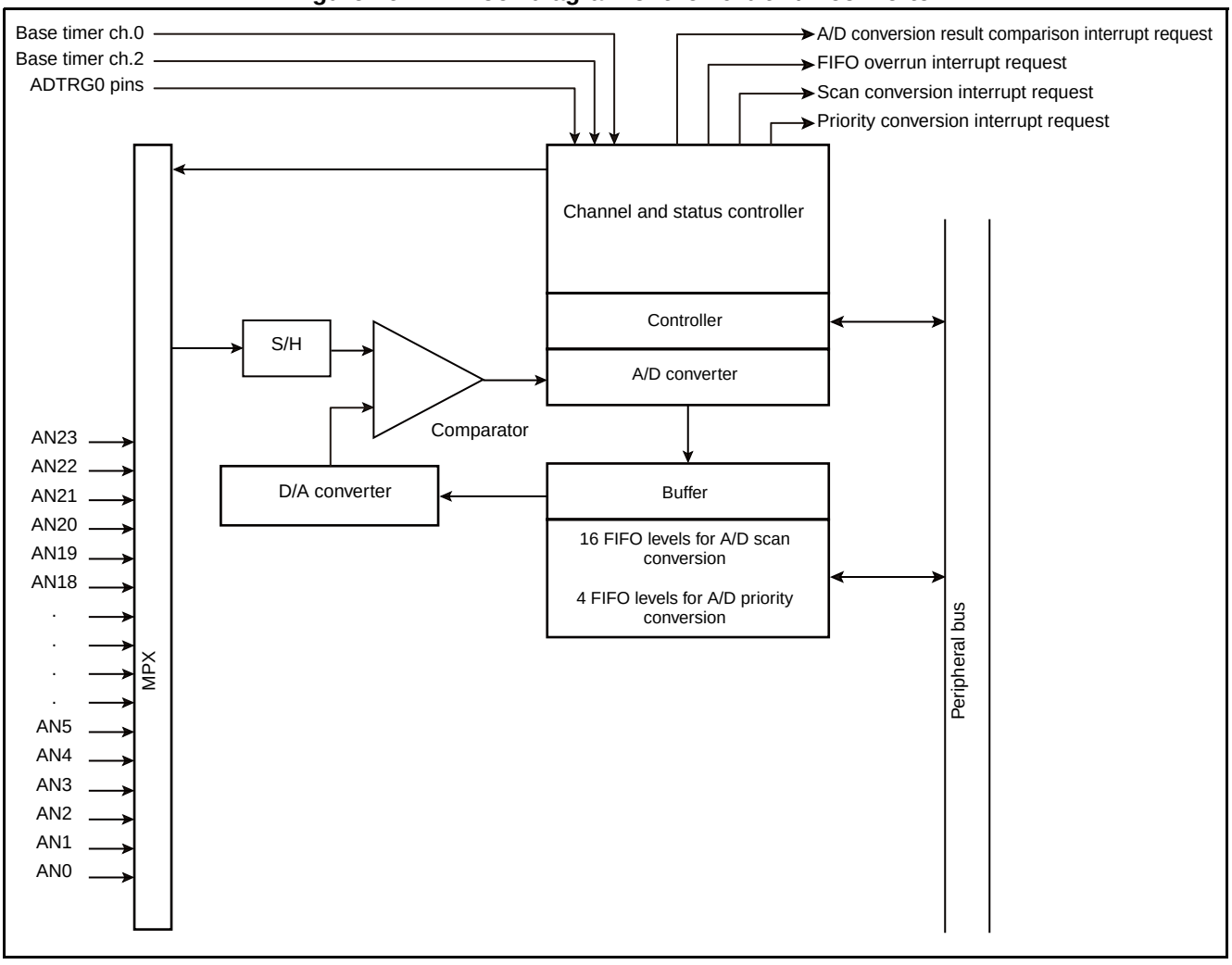
26.2 Configuration

This section explains the configuration of the 10-bit A/D converter.

■ Block diagram of the 10-bit A/D converter

Figure 26.2-1 shows a block diagram of the 10-bit A/D converter.

Figure 26.2-1 Block diagram of the 10-bit A/D converter



- A/D scan conversion FIFO
  This is the FIFO for A/D scan conversion. There are 16 FIFO levels.
- A/D priority conversion FIFO
  This is the FIFO for A/D priority conversion. There are four FIFO levels.
- Controller
  This controller controls conversion operations.
- Channel and status controller
  This controller controls the channels and status of the 10-bit A/D converter.
- MPX (analog multiplexer)
  The MPX selects (switches to), from multiple analog input signals, the analog signal to be converted.

■ Clocks

Table 26.2-1 lists the clocks used for the 10-bit A/D converter.

Table 26.2-1 Clock used for the 10-bit A/D converter

Clock Name	Description
Operation clock	Peripheral clock (PCLK)

26.3 Pins

This section explains the pins used for the 10-bit A/D converter.

■ Overview

The 10-bit A/D converter has the following pins:

- AV_{CC} pin

  10-bit A/D converter analog power input pin
- AVRH pin

  10-bit A/D converter reference voltage input pin
- AV_{SS} pin

  10-bit A/D converter GND pin
- AN0 to AN23 pins

  10-bit A/D converter analog input pins

  These pins are multiplexed pins. For details of using these pins as the AN0 to AN23 pins of the 10-bit A/D converter, see "15.4.6 A/D Channel Enable Register (ADCHE)".
- ADTRG0 pins

  10-bit A/D converter external trigger input pins

  These pins are multiplexed pins. For details of using these pins as the ADTRG0 pins of the 10-bit A/D converter, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 26.3-1 shows the relationship between channels and pins.

Table 26.3-1 Relationship between channels and pins

Channel	Analog Power Input Pin	Reference Voltage Input Pin	GND Pin	Analog Input Pin	External Trigger Input Pin
0	AV_{CC}	AVRH	AV_{SS}	AN0	ADTRG0
1				AN1	
2				AN2	
3				AN3	
4				AN4	
5				AN5	
6				AN6	
7				AN7	
8				AN8	
9				AN9	
10				AN10	
11				AN11	
12				AN12	
13				AN13	
14				AN14	
15				AN15	
16				AN16	
17				AN17	
18				AN18	
19				AN19	
20				AN20	
21				AN21	
22				AN22	
23				AN23	

26.4 Registers

This section explains the configurations and functions of the registers used for the 10-bit A/D converter.

■ List of registers for the 10-bit A/D converter

Table 26.4-1 lists the registers used for the 10-bit A/D converter.

Table 26.4-1 Registers for the 10-bit A/D converter

Abbreviated Register Name	Register Name	Reference
ADCHE	A/D channel enable register	15.4.6
ADCR0	A/DC control register 0	26.4.1
ADSR0	A/DC status register 0	26.4.2
SCCR0	Scan conversion control register 0	26.4.3
SFNS0	Scan conversion FIFO number setting register 0	26.4.4
SCIS00	Scan conversion input select register 00	26.4.6
SCIS10	Scan conversion input select register 10	26.4.6
SCIS20	Scan conversion input select register 20	26.4.6
SCFD0	Scan conversion FIFO data register 0	26.4.5
PCCR0	Priority conversion control register 0	26.4.7
PFNS0	Priority conversion FIFO number setting register 0	26.4.8
PCIS0	Priority conversion input select register 0	26.4.10
PCFD0	Priority conversion FIFO data register 0	26.4.9
CMPD0	A/D comparison data setting register 0	26.4.11
CMPCR0	A/D comparison control register 0	26.4.12
ADSS00	Sampling time select register 00	26.4.14
ADSS10	Sampling time select register 10	26.4.14
ADSS20	Sampling time select register 20	26.4.14
ADST00	Sampling time setting register 00	26.4.13
ADST10	Sampling time setting register 10	26.4.13
ADCT0	Compare time setting register 0	26.4.15

26.4.1 A/DC Control Registers (ADCR0)

These registers control interrupt requests.

Figure 26.4-1 shows the bit configuration of the A/DC control registers (ADCR0).

Figure 26.4-1 Bit configuration of the A/DC control registers (ADCR0)

bit	7	6	5	4	3	2	1	0
	SCIF	PCIF	CMPIF	Undefined	SCIE	PCIE	CMPIE	OVRIE
Attribute	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Initial value	0	0	0	X	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7]: SCIF (Scan conversion interrupt request flag bit)

This bit indicates that A/D scan conversion results have been stored in the number of stages in the FIFO as specified by the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

If the SCIE bit is set to "1" when this bit is "1", a scan conversion interrupt request is generated.

SCIF	In Case of Reading	In Case of Writing
0	The number of stages storing conversion results has not reached the specified number of stages.	This bit is cleared to "0".
1	The number of stages storing conversion results has reached the specified number of stages.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit6]: PCIF (Priority conversion interrupt request flag bit)

This bit indicates that A/D priority conversion results have been stored up to the number of stages in the FIFO as specified by the PFS1 and PFS0 bits in a priority conversion FIFO number setting register (PFNS0).

If the PCIE bit is set to "1" when this bit is "1", a priority conversion interrupt request is generated.

PCIF	In Case of Reading	In Case of Writing
0	The number of stages storing conversion results has not reached the specified number of stages.	This bit is cleared to "0".
1	The number of stages storing conversion results has reached the specified number of stages.	Ignored

---

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit5]: CMPIF (Conversion result comparison interrupt request flag bit)

The A/D conversion result compare function is used when conversion results are compared with the data in the A/D comparison data setting registers (CMPD0).

This bit indicates that a conversion result satisfies the requirements set in an A/D comparison data setting register (CMPD0) and an A/D comparison control register (CMPCR0).

If the CMPIE bit is set to "1" when this bit is "1", a conversion result comparison interrupt request is generated.

CMPIF	In Case of Reading	In Case of Writing
0	The requirements are not satisfied.	This bit is cleared to "0".
1	The requirements are satisfied.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit4]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit3]: SCIE (Scan conversion interrupt enable bit)

This bit specifies whether to generate a scan conversion interrupt request when the number of stages storing A/D scan conversion results reaches the number of FIFO stages (SCIF bit = 1) specified in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

Written Value	Explanation
0	Disables generation of scan conversion interrupt requests.
1	Enables generation of scan conversion interrupt requests.

[bit2]: PCIE (Priority conversion interrupt enable bit)

This bit specifies whether to generate a priority conversion interrupt request when the number of stages storing A/D priority conversion results reaches the number of FIFO stages (PCIF bit = 1) specified in the PFS1 and PFS0 bits in a priority conversion FIFO number setting register (PFNS0).

Written Value	Explanation
0	Disables generation of priority conversion interrupt requests.
1	Enables generation of priority conversion interrupt requests.

[bit1]: CMPIE (Conversion result comparison interrupt enable bit)

The A/D conversion result compare function is used when conversion results are compared with the data in the A/D comparison data setting registers (CMPD0).

This bit specifies whether to generate a conversion result comparison interrupt request when a conversion result satisfies the requirements (CMPIF bit = 1) set in an A/D comparison control register (CMPCR0).

Written Value	Explanation
0	Disables generation of conversion result comparison interrupt requests.
1	Enables generation of conversion result comparison interrupt requests.

[bit0]: OVRIE (FIFO overrun interrupt enable bit)

This bit specifies whether to generate a FIFO overrun interrupt request when the SOVR bit in a scan conversion control register (SCCR0) or the POVR bit in a priority conversion control register (PCCR0) changes to "1".

If an attempt is made to write to a full FIFO, the SOVR bit in the scan conversion control register (SCCR0) or the POVR bit in the priority conversion control register (PCCR0) changes to "1".

Written Value	Explanation
0	Disables generation of FIFO overrun interrupt requests.
1	Enables generation of FIFO overrun interrupt requests.

26.4.2 A/DC Status Registers (ADSR0)

These registers indicate the A/D conversion status.

Figure 26.4-2 shows the bit configuration of the A/DC status registers (ADSR0).

Figure 26.4-2 Bit configuration of the A/DC status registers (ADSR0)

bit	7	6	5	4	3	2	1	0
	ADSTP	FDAS	Undefined	Undefined	Undefined	PCNS	PCS	SCS
Attribute	R/W	R/W	-	-	-	R	R	R
Initial value	0	0	X	X	X	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit7]: ADSTP (A/D conversion abort bit)

This bit forcibly stops A/D conversion.

ADSTP	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Forcibly stops A/D conversion.	

<Notes>

- Writing "1" to this bit stops A/D conversion in either A/D scan conversion mode or A/D priority conversion mode.
- Writing "1" to this bit to forcibly stop A/D conversion clears the PCNS, PCS, and SCS bits to "0". However, it does not affect other registers.

[bit6]: FDAS (FIFO data allocation select bit)

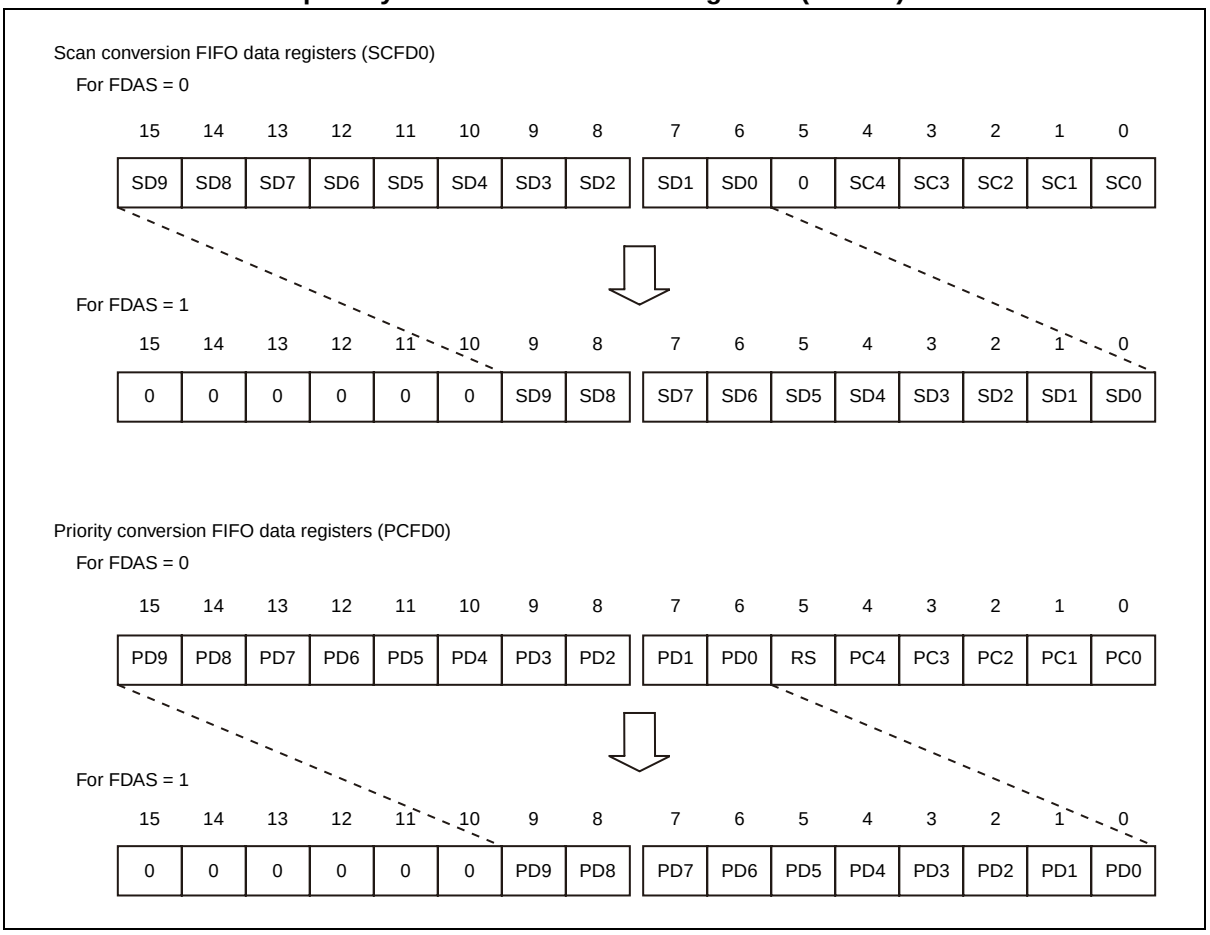
This bit specifies the mode of bit allocation to the scan conversion FIFO data registers (SCFD0) and priority conversion FIFO data registers (PCFD0).

- Left-justify: Conversion results (with channel information) are left-justified.
- Right-justify: Conversion results (without channel information) are shifted 6 bits to the LSB side to right-justify the results. Conversion results are allocated to bit9 to bit0.

Written Value	Explanation
0	Allocates conversion results left-justified.
1	Allocates conversion results right-justified.

Figure 26.4-3 shows the relationship between this bit and the scan conversion FIFO data registers (SCFD0) and the relationship between this bit and the priority conversion FIFO data registers (PCFD0).

Figure 26.4-3 Relationship between FDAS and the scan conversion FIFO data registers (SCFD0)/ priority conversion FIFO data registers (PCFD0)


Scan conversion FIFO data registers (SCFD0)
For FDAS = 0
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
SD9 SD8 SD7 SD6 SD5 SD4 SD3 SD2 SD1 SD0 0 SC4 SC3 SC2 SC1 SC0
For FDAS = 1
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 SD9 SD8 SD7 SD6 SD5 SD4 SD3 SD2 SD1 SD0
Priority conversion FIFO data registers (PCFD0)
For FDAS = 0
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
PD9 PD8 PD7 PD6 PD5 PD4 PD3 PD2 PD1 PD0 RS PC4 PC3 PC2 PC1 PC0
For FDAS = 1
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 0 0 0 0 0 PD9 PD8 PD7 PD6 PD5 PD4 PD3 PD2 PD1 PD0

<Notes>

- If "1" is written to this bit to select right-justification, conversion results are shifted six bits to the LSB side, which consequently leads to a loss of converted information on channels (the SC4 to SC0 bits/PC4 to PC0 bits in Figure 26.4-3). Right-justification is used only when channel information is not required in conversion results, such as when conversion involves only 1 channel.
- If "1" is written to this bit to select right-justification in A/D priority conversion mode, (the RS bit of) activation trigger information on A/D priority conversion is lost. Right-justification is used only when either priority 1 or 2 of A/D priority conversion mode is used.

[bit5 to bit3]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2]: PCNS (Priority conversion pending flag bit)

This bit indicates that A/D priority conversion of priority 2 is pending.

If A/D priority conversion of priority 2 is activated during execution of A/D priority conversion of priority 1 or vice versa, the bit is changed to "1".

Read Value	Explanation
0	A/D priority conversion of priority 2 is not pending.
1	A/D priority conversion of priority 2 is pending.

[bit1]: PCS (Priority conversion status flag bit)

This bit indicates that A/D priority conversion of priority 1 or 2 is in progress.

Read Value	Explanation
0	A/D priority conversion is stopped.
1	A/D priority conversion is in progress.

[bit0]: SCS (Scan conversion status flag bit)

This bit indicates that A/D scan conversion is in progress.

Read Value	Explanation
0	A/D scan conversion is stopped.
1	A/D scan conversion is in progress.

26.4.3 Scan Conversion Control Registers (SCCR0)

These registers are used to control the operation of A/D scan conversion.

Figure 26.4-4 shows the bit configuration of the scan conversion control registers (SCCR0).

Figure 26.4-4 Bit configuration of the scan conversion control registers (SCCR0)

bit	7	6	5	4	3	2	1	0
	SEMP	SFUL	SOVR	SFCLR	Undefined	RPT	SHEN	SSTR
Attribute	R	R	R/W	R/W	-	R/W	R/W	R/W
Initial value	1	0	0	0	X	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

<Note>

Byte access to these registers must be performed independently, or half word access must be performed with a scan conversion FIFO number setting register (SFNS0).

The scan conversion FIFO data register (SCFD0) needs to be read when the SEMP bit is "0".

[bit7]: SEMP (Scan conversion FIFO empty flag bit)

This bit indicates that the FIFO for A/D scan conversion is empty.

Read Value	Explanation
0	The FIFO for A/D scan conversion contains data.
1	The FIFO for A/D scan conversion is empty.

This bit is cleared to "0" when data is stored in a scan conversion FIFO data register (SCFD0).

[bit6]: SFUL (Scan conversion FIFO full bit)

This bit indicates that the FIFO for A/D scan conversion is full.

Read Value	Explanation
0	The FIFO for A/D scan conversion has free space.
1	The FIFO for A/D scan conversion is full.

This bit is cleared to "0" when "1" is written to the SFCLR bit or a scan conversion FIFO data register (SCFD0) is read.

[bit5]: SOVR (Scan conversion overrun flag bit)

The bit indicates that an attempt has been made to write to a full A/D scan conversion FIFO (an overrun has occurred).

If the OVRIE bit in an A/DC control register (ADCR0) is set to "1" when this bit is "1", a FIFO overrun interrupt request is generated.

SOVR	In Case of Reading	In Case of Writing
0	No overrun occurred.	This bit is cleared to "0".
1	An overrun occurred.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- When an attempt is made to write data to a full FIFO, the conversion data in the FIFO is not overwritten.

[bit4]: SFCLR (Scan conversion FIFO clear bit)

This bit is used to clear the A/D scan conversion FIFO.

SFCLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Clears the A/D scan conversion FIFO.	

<Note>

Writing "1" to this bit empties the FIFO for A/D scan conversion. Accordingly, the SEMP bit changes to "1".

[bit3]: Undefined bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2]: RPT (Scan conversion repeat bit)

This bit specifies the A/D scan conversion mode.

- Single conversion mode: Signals from the channel specified by a scan conversion input select register (SCIS20 to SCIS00) are converted only once.
- Repeat conversion mode: Signals from the channel specified by a scan conversion input select register (SCIS20 to SCIS00) are converted repeatedly.

Written Value	Explanation
0	Single conversion mode
1	Repeat conversion mode

<Notes>

- If "0" is written to this bit during conversion in repeat conversion mode, the conversion operation is stopped after the signals from the channel specified by the scan conversion input select register (SCIS20 to SCIS00) are converted.
- To enable repeat conversion mode, write "1" to this bit after checking the SCS bit in the A/DC status registers (ADSR0) and confirming that A/D scan conversion is stopped (SCS = 0).

  However, to start A/D scan conversion (with SSTR = 1) while simultaneously enabling repeat conversion mode, the SSTR bit can be written at the same time as this bit.

[bit1]: SHEN (Scan conversion timer activation enable bit)

This bit specifies whether to activate A/D scan conversion upon detection of the rising edge of a TOUT signal of base timer ch.0.

Written Value	Explanation
0	Disables A/D scan conversion activation based on a base timer (ch.0).
1	Enables A/D scan conversion activation based on a base timer (ch.0).

<Notes>

- If "1" is written to the SSTR bit, A/D scan conversion is activated regardless of the setting of this bit.
- After "1" is written to this bit, "1" may be written to the SSTR bit at the same time that activation is triggered by a base timer (ch.0). In this event, activation by software is given priority, and activation triggered by the base timer is ignored.
- For details of the TOUT signal, see "CHAPTER 24 Base Timer".

[bit0]: SSTR (Scan conversion start bit)

This bit is used to activate A/D scan conversion by software.

Writing "1" to the bit during conversion stops and restarts the conversion.

SSTR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates or reactivates A/D scan conversion.	

26.4.4 Scan Conversion FIFO Number Setting Register (SFNS0)

These registers specify the maximum number of stages in the A/D scan conversion FIFO. A scan conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D scan conversion.

Figure 26.4-5 shows the bit configuration of the scan conversion FIFO number setting registers (SFNS0).

Figure 26.4-5 Bit configuration of the scan conversion FIFO number setting registers (SFNS0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	SFS3	SFS2	SFS1	SFS0
Attribute	-	-	-	-	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

<Note>

Byte access to these registers must be performed independently, or half word access must be performed with a scan conversion control register (SCCR0).

[bit7 to bit4]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit3 to bit0]: SFS3 to SFS0 (Scan conversion FIFO number setting bits)

These bits specify the maximum number of stages in the A/D scan conversion FIFO. A scan conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D scan conversion.

The SCIF bit in an A/DC control register (ADCR0) changes to "1" when the number of stages storing such data reaches the maximum number of FIFO stages specified by these bits.

SFS3	SFS2	SFS1	SFS0	Explanation
0	0	0	0	First level
0	0	0	1	Second level
0	0	1	0	Third level
0	0	1	1	Fourth level
0	1	0	0	Fifth level
0	1	0	1	Sixth level
0	1	1	0	Seventh level
0	1	1	1	Eighth level
1	0	0	0	Ninth level
1	0	0	1	Tenth level
1	0	1	0	Eleventh level
1	0	1	1	Twelfth level
1	1	0	0	Thirteenth level
1	1	0	1	Fourteenth level
1	1	1	0	Fifteenth level
1	1	1	1	Sixteenth level

26.4.5 Scan Conversion FIFO Data Registers (SCFD0)

These registers store A/D scan conversion results. Each register consists of 16 FIFO stages.
FIFO data can be read sequentially from the registers.
The bit configuration of these registers varies depending on the setting of the FDAS bit in the A/DC status registers (ADSR0).

<Notes>

- One of these registers must always be read after the SEMP bit in a scan conversion control register (SCCR0) is checked to determine whether data remains in the A/D scan conversion FIFO (SEMP = 0).

  If this register is read when the A/D scan conversion FIFO is empty (SEMP = 1), it is impossible to determine whether the read data is valid. For details, see "■ Operation of A/D scan conversion" in "26.6.3 FIFO Operations".
- Do not perform word access to these registers.
- In byte access to these registers, the low-order byte (bit7 to bit0) must be accessed before the high-order byte (bit15 to bit8). FIFO data is shifted after the high-order bytes is read.

■ Left-justify (FDAS = 0)

Figure 26.4-6 shows the bit configuration of the scan conversion FIFO data registers (SCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies left-justification (FDAS = 0).

Figure 26.4-6 Bit configuration of the scan conversion FIFO data registers (SCFD0)

bit	15	14	13	12	11	10	9	8
	SD9	SD8	SD7	SD6	SD5	SD4	SD3	SD2
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	SD1	SD0	Undefined	SC4	SC3	SC2	SC1	SC0
Attribute	R	R	-	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit6]: SD9 to SD0 (A/D scan conversion result bits)

These bits store A/D scan conversion results.

[bit5]: Undefined bit

"0" is read.

[bit4 to bit0]: SC4 to SC0 (Conversion channel bits)

These bits indicate the channel from which analog input has been converted and stored in the SD9 to SD0 bits.

SC4	SC3	SC2	SC1	SC0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	ch.8 (AN8 pin)
0	1	0	0	1	ch.9 (AN9 pin)
0	1	0	1	0	ch.10 (AN10 pin)
0	1	0	1	1	ch.11 (AN11 pin)
0	1	1	0	0	ch.12 (AN12 pin)
0	1	1	0	1	ch.13 (AN13 pin)
0	1	1	1	0	ch.14 (AN14 pin)
0	1	1	1	1	ch.15 (AN15 pin)
1	0	0	0	0	ch.16 (AN16 pin)
1	0	0	0	1	ch.17 (AN17 pin)
1	0	0	1	0	ch.18 (AN18 pin)
1	0	0	1	1	ch.19 (AN19 pin)
1	0	1	0	0	ch.20 (AN20 pin)
1	0	1	0	1	ch.21 (AN21 pin)
1	0	1	1	0	ch.22 (AN22 pin)
1	0	1	1	1	ch.23 (AN23 pin)
1	1	0	0	0	Setting prohibited
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	

SC4	SC3	SC2	SC1	SC0	Explanation
1	1	1	0	0	Setting prohibited
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

■ Right-justify (FDAS = 1)

Figure 26.4-7 shows the bit configuration of the scan conversion FIFO data registers (SCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies right-justification (FDAS = 1).

Figure 26.4-7 Bit configuration of the scan conversion FIFO data registers (SCFD0)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	SD9	SD8
Attribute	-	-	-	-	-	-	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	SD7	SD6	SD5	SD4	SD3	SD2	SD1	SD0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit10]: Undefined bits

"0" is read.

[bit9 to bit0]: SD9 to SD0 (A/D scan conversion result bits)

These bits store A/D scan conversion results.

<Note>

Information on converted channels is not stored in right-justify mode. Right-justification is used only when channel information is not required in conversion results, such as when conversion involves only one channel.

26.4.6 Scan Conversion Input Select Registers (SCIS20 to SCIS00)

These registers are used to select the channel to be subject to A/D scan conversion.

Figure 26.4-8 shows the bit configuration of the scan conversion input select registers (SCIS20 to SCIS00).

Figure 26.4-8 Bit configuration of the scan conversion input select registers (SCIS20 to SCIS00)

Scan conversion input select register 20 (SCIS20)

bit	7	6	5	4	3	2	1	0
	AN23	AN22	AN21	AN20	AN19	AN18	AN17	AN16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Scan conversion input select register 10 (SCIS10)

bit	7	6	5	4	3	2	1	0
	AN15	AN14	AN13	AN12	AN11	AN10	AN9	AN8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Scan conversion input select register 00 (SCIS00)

bit	7	6	5	4	3	2	1	0
	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

SCIS20 to SCIS00: AN23 to AN0 (analog input select bits)

The channel corresponding to a bit that is set to "1" is made subject to conversion.

The AN23 bit corresponds to ch.23 (AN23 pin), the AN22 bit corresponds to ch.22 (AN22 pin), ... the AN1 bit corresponds to ch.1 (AN1 pin), and the AN0 bit corresponds to ch.0 (AN0 pin).

If multiple channels are selected with these registers, they are made subject to conversion sequentially in ascending order of channel number. For example, if "1" is written to the AN3, AN5, AN10, and AN23 bits, the corresponding channels are made subject to conversion in the following sequence:

ch.3 → ch.5 → ch.10 → ch.23

<Note>

Write to these registers while A/D conversion is stopped.

26.4.7 Priority Conversion Control Registers (PCCR0)

These registers are used to control the operation of A/D priority conversion. Two priority levels can be selected.

Figure 26.4-9 shows the bit configuration of the priority conversion control registers (PCCR0).

Figure 26.4-9 Bit configuration of the priority conversion control registers (PCCR0)

bit	7	6	5	4	3	2	1	0
	PEMP	PFUL	POVR	PFCLR	Reserved	PEEN	PHEN	PSTR
Attribute	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

<Note>

Byte access to these registers must be performed independently, or half word access must be performed with a priority conversion FIFO number setting register (PFNS0).

The priority conversion FIFO data register (PCFD0) needs to be read when the SEMP bit is "0".

[bit7]: PEMP (Priority conversion FIFO empty flag bit)

This bit indicates that the FIFO for A/D priority conversion is empty.

Read Value	Explanation
0	The FIFO for A/D priority conversion contains data.
1	The FIFO for A/D priority conversion is empty.

This bit is cleared to "0" when data is stored in the priority conversion FIFO data register (PCFD0).

[bit6]: PFUL (Priority conversion FIFO full bit)

This bit indicates that the FIFO for A/D priority conversion is full.

Read Value	Explanation
0	The A/D priority conversion FIFO has free space.
1	The A/D priority conversion FIFO is full.

This bit is cleared to "0" when "1" is written to the PFCLR bit or a priority conversion FIFO data register (PCFD0) is read.

[bit5]: POVR (Priority conversion overrun flag bit)

This bit indicates that an attempt has been made to write to a full A/D priority conversion FIFO (an overrun has been occurred).

If the OVRIE bit in an A/DC control register (ADCR0) is set to "1" when this bit is "1", a FIFO overrun interrupt request is generated.

POVR	In Case of Reading	In Case of Writing
0	Overrun has not been occurred	This bit is cleared to "0".
1	Overrun has been occurred	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- Even if an attempt is made to write data to a full FIFO, the conversion data in the FIFO is not overwritten.

[bit4]: PFCLR (Priority conversion FIFO clear bit)

This bit is used to clear the A/D priority conversion FIFO.

PFCLR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Clears the A/D priority conversion FIFO.	

<Note>

Writing "1" to this bit empties the FIFO for A/D priority conversion. Accordingly, the PEMP bit changes to "1".

[bit3]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2]: PEEN (Priority conversion external activation enable bit)

This bit specifies whether to activate A/D priority conversion of priority level 1 upon detection of a falling edge from the ADTRG0 pin. Priority 1 has the highest priority because priority 1 > priority 2.

Written Value	Explanation
0	Disables activation of A/D priority conversion of priority 1.
1	Enables activation of A/D priority conversion of priority 1.

<Note>

The microcontroller provides four pins that can be used as an ADTRG0 pin. Specify a pin as the ADTRG0 pin.

For details of how to set the pin, see "CHAPTER 15 I/O Ports".

[bit1]: PHEN (Priority conversion timer activation enable bit)

This bit specifies whether to activate A/D priority conversion of priority 2 upon detection of the rising edge of a TOUT signal of base timer ch.2. Priority 2 < priority 1.

Written Value	Explanation
0	Disables activation of A/D priority conversion of priority 2.
1	Enables activation of A/D priority conversion of priority 2.

<Notes>

- If "1" is written to the PSTR bit, A/D priority conversion of priority 2 is activated regardless of the setting of this bit.
- For details of the TOUT signal, see "CHAPTER 24 Base Timer".

[bit0]: PSTR (Priority conversion start bit)

This bit enables software to activate A/D priority conversion of priority 2. Priority 2 < priority 1.

PSTR	In Case of Writing	In Case of Reading
0	Ignored	"0" is read.
1	Activates A/D priority conversion of priority 2.	

<Note>

Even if "1" is written to this bit during A/D conversion, the A/D conversion cannot be reactivated.

26.4.8 Priority Conversion FIFO Number Setting Registers (PFNS0)

These registers specify the maximum number of stages in the A/D priority conversion FIFO. A priority conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D priority conversion.

Figure 26.4-10 shows the bit configuration of the priority conversion FIFO number setting registers (PFNS0).

Figure 26.4-10 Bit configuration of the Priority conversion FIFO number setting registers (PFNS0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	PFS1	PFS0
Attribute	-	-	-	-	-	-	R/W	R/W
Initial value	X	X	X	X	X	X	0	0

R/W: Read/Write

-: Undefined

X: Undefined

<Note>

Byte access to these registers must be performed independently, or half word access must be performed with a priority conversion control register (PCCR0).

The priority conversion FIFO data register (PCFD0) needs to be read when the PEMP bit is "0".

[bit7 to bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1, bit0]: PFS1, PFS0 (Priority conversion FIFO number setting bits)

These bits specify the maximum number of stages in the A/D priority conversion FIFO. A priority conversion interrupt request can be issued when the number of stages storing conversion results reaches that maximum number during A/D priority conversion.

The PCIF bit in an A/DC control register (ADCR0) changes to "1" when the number of stages storing such data reaches the maximum number of FIFO stages specified by these bits.

PFS1	PFS0	Explanation
0	0	First level
0	1	Second level
1	0	Third level
1	1	Fourth level

26.4.9 Priority Conversion FIFO Data Registers (PCFD0)

These registers store A/D priority conversion results. Each register consists of 4 FIFO stages. FIFO data can be read sequentially from the registers.
The bit configuration of these registers varies depending on the setting of the FDAS bit in the A/DC status registers (ADSR0).

<Notes>

- One of these registers must always be read after the PEMP bit in a priority conversion control register (PCCR0) is checked to determine whether data remains in the A/D priority conversion FIFO (PEMP = 0).

  If this register is read when the A/D priority conversion FIFO is empty (PEMP = 1), it is impossible to determine whether the read data is valid. For details, see "■ Operation of A/D priority conversion" in "26.6.3 FIFO Operations".
- Do not perform word access to these registers.
- In byte access to these registers, the low-order byte (bit7 to bit0) must be accessed before the high-order byte (bit15 to bit8). FIFO data is shifted after the high-order byte is read.

■ Left-justify (FDAS = 0)

Figure 26.4-11 shows the bit configuration of the priority conversion FIFO data registers (PCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies left-justification (FDAS = 0).

Figure 26.4-11 Bit configuration of the priority conversion FIFO data registers (PCFD0)

bit	15	14	13	12	11	10	9	8
	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	PD1	PD0	RS	PC4	PC3	PC2	PC1	PC0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

X: Undefined

[bit15 to bit6]: PD9 to PD0 (A/D priority conversion result bits)

These bits store A/D priority conversion results.

[bit5]: RS (Priority A/D activation trigger bit)

This bit indicates whether the data stored in the PD9 to PD0 bits has been converted with priority 1 or 2 (activation trigger for A/D priority conversion).

Read Value	Explanation
0	Priority 2 (activation by software/base timer)
1	Priority 1 (activation by an external trigger)

<Note>

The activation trigger for A/D priority conversion of priority 2 cannot be distinguished as software or a base timer.

[bit4 to bit0]: PC4 to PC0 (Conversion channel bits)

These bits indicate the channel from which analog input has been converted and stored in the PD9 to PD0 bits.

PC4	PC3	PC2	PC1	PC0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	ch.8 (AN8 pin)
0	1	0	0	1	ch.9 (AN9 pin)
0	1	0	1	0	ch.10 (AN10 pin)
0	1	0	1	1	ch.11 (AN11 pin)
0	1	1	0	0	ch.12 (AN12 pin)
0	1	1	0	1	ch.13 (AN13 pin)
0	1	1	1	0	ch.14 (AN14 pin)
0	1	1	1	1	ch.15 (AN15 pin)
1	0	0	0	0	ch.16 (AN16 pin)
1	0	0	0	1	ch.17 (AN17 pin)
1	0	0	1	0	ch.18 (AN18 pin)

PC4	PC3	PC2	PC1	PC0	Explanation
1	0	0	1	1	ch.19 (AN19 pin)
1	0	1	0	0	ch.20 (AN20 pin)
1	0	1	0	1	ch.21 (AN21 pin)
1	0	1	1	0	ch.22 (AN22 pin)
1	0	1	1	1	ch.23 (AN23 pin)
1	1	0	0	0	Setting prohibited
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

<Note>

A/D priority conversion of priority 1 can be performed only for ch.0 to ch.7.

■ Right-justify (FDAS = 1)

Figure 26.4-12 shows the bit configuration of the priority conversion FIFO data registers (PCFD0) when the FDAS bit in an A/DC status register (ADSR0) specifies right-justification (FDAS = 1).

Figure 26.4-12 Bit configuration of the Priority conversion FIFO data registers (PCFD0)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	PD9	PD8
Attribute	-	-	-	-	-	-	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

R: Read only

-: Undefined

X: Undefined

[bit15 to bit10]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit9 to bit0]: PD9 to PD0 (A/D priority conversion result bits)

These bits store A/D priority conversion results.

<Note>

The activation trigger (priority) for A/D priority conversion and information on the converted channel are not stored in right-justify mode. Right-justification is used only when only either priority 1 or 2 of A/D priority conversion mode is used and when the converted result does not need the channel information, such as a conversion with 1 channel.

26.4.10 Priority Conversion Input Select Registers (PCIS0)

These registers are used to select the channel to be subject to A/D priority conversion.
1 channel subject to conversion with priority 2 is selected from the 24 channels, and 1 channel subject to conversion with priority 1 is selected from ch.0 to ch.7.

Figure 26.4-13 shows the bit configuration of the priority conversion input select registers (PCIS0).

Figure 26.4-13 Bit configuration of the priority conversion input select registers (PCIS0)

bit	7	6	5	4	3	2	1	0
	P2A4	P2A3	P2A2	P2A1	P2A0	P1A2	P1A1	P1A0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7 to bit3]: P2A4 to P2A0 (Priority 2 analog input select bit)

These bits select the channel to be subject to A/D priority conversion of priority 2. Priority 2 < priority 1.

P2A4	P2A3	P2A2	P2A1	P2A0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	ch.8 (AN8 pin)
0	1	0	0	1	ch.9 (AN9 pin)
0	1	0	1	0	ch.10 (AN10 pin)
0	1	0	1	1	ch.11 (AN11 pin)
0	1	1	0	0	ch.12 (AN12 pin)
0	1	1	0	1	ch.13 (AN13 pin)
0	1	1	1	0	ch.14 (AN14 pin)
0	1	1	1	1	ch.15 (AN15 pin)
1	0	0	0	0	ch.16 (AN16 pin)
1	0	0	0	1	ch.17 (AN17 pin)
1	0	0	1	0	ch.18 (AN18 pin)
1	0	0	1	1	ch.19 (AN19 pin)
1	0	1	0	0	ch.20 (AN20 pin)
1	0	1	0	1	ch.21 (AN21 pin)
1	0	1	1	0	ch.22 (AN22 pin)
1	0	1	1	1	ch.23 (AN23 pin)
1	1	0	0	0	Setting prohibited
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

[bit2 to bit0]: P1A2 to P1A0 (Priority 1 analog input select bit)

These bits select the channel to be subject to A/D priority conversion of priority 1. A/D priority conversion of priority 1 can be performed only for ch.0 to ch.7. Priority 2 < priority 1.

P1A2	P1A1	P1A0	Explanation
0	0	0	ch.0 (AN0 pin)
0	0	1	ch.1 (AN1 pin)
0	1	0	ch.2 (AN2 pin)
0	1	1	ch.3 (AN3 pin)
1	0	0	ch.4 (AN4 pin)
1	0	1	ch.5 (AN5 pin)
1	1	0	ch.6 (AN6 pin)
1	1	1	ch.7 (AN7 pin)

26.4.11 A/D Comparison Data Setting Registers (CMPD0)

These registers are used to set the value that is compared with A/D conversion results when the comparison function is used. The eight high-order bits of the conversion results are compared with a value set in the registers. If the comparison result satisfies the requirements set in an A/D comparison control register (CMPCR0), the CMPIF bit in an A/DC control register (ADCR0) changes to "1".

Figure 26.4-14 shows the bit configuration of the A/D comparison data setting registers (CMPD0).

Figure 26.4-14 Bit configuration of the A/D comparison data setting registers (CMPD0)

bit	7	6	5	4	3	2	1	0
	CMAD9	CMAD8	CMAD7	CMAD6	CMAD5	CMAD4	CMAD3	CMAD2
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

A value that is set in these registers is compared with the eight high-order bits (bit9 to bit2) of A/D conversion results. The two bits (bit1, bit0) on the LSB side of A/D conversion results are not used for the comparison.

26.4.12 A/D Comparison Control Registers (CMPCR0)

These registers control the comparison function. The comparison function is used when an A/D conversion result is compared with the value set in an A/D comparison data setting register (CMPD0). If the comparison result satisfies the requirements set in that register, the CMPIF bit in an A/DC control register (ADCR0) changes to "1".

Figure 26.4-15 shows the bit configuration of the A/D comparison control registers (CMPCR0).

Figure 26.4-15 Bit configuration of the A/D comparison control registers (CMPCR0)

bit	7	6	5	4	3	2	1	0
	CMPEN	CMD1	CMD0	CCH4	CCH3	CCH2	CCH1	CCH0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

[bit7]: CMPEN (Comparison function operation enable bit)

This bit specifies whether to enable the comparison function.

Written Value	Explanation
0	Disables the comparison function.
1	Enables the comparison function.

[bit6]: CMD1 (Comparison mode 1 bit)

This bit sets the conversion interrupt request generation conditions.

Written Value	Explanation
0	A conversion result interrupt request is generated when the A/D conversion result is smaller than the value set in an A/D comparison data setting register (CMPD0).
1	A conversion result interrupt request is generated when the A/D conversion result is equal to or greater than the value set in an A/D comparison data setting register (CMPD0).

[bit5]: CMD0 (Comparison mode 0 bit)

This bit selects one of the following comparison modes:

- Comparing the conversion result of the channel specified by the CCH4 to CCH0 bits with the value set in an A/D comparison data setting register (CMPD0)
- Comparing the conversion results of all channels with the value set in an A/D comparison data setting register (CMPD0)

Written Value	Explanation
0	Compares the conversion result of the channel specified by the CCH4 to CCH0 bits.
1	Compares the conversion results of all channels.

<Note>

Writing "1" to this bit invalidates the settings of the CCH4 to CCH0 bits.

[bit4 to bit0]: CCH4 to CCH0 (Comparison target analog input channel bits)

These bits specify the channel to be compared with the value set in an A/D comparison data setting register (CMPD0) when the CMD0 bit is "0".

CCH4	CCH3	CCH2	CCH1	CCH0	Explanation
0	0	0	0	0	ch.0 (AN0 pin)
0	0	0	0	1	ch.1 (AN1 pin)
0	0	0	1	0	ch.2 (AN2 pin)
0	0	0	1	1	ch.3 (AN3 pin)
0	0	1	0	0	ch.4 (AN4 pin)
0	0	1	0	1	ch.5 (AN5 pin)
0	0	1	1	0	ch.6 (AN6 pin)
0	0	1	1	1	ch.7 (AN7 pin)
0	1	0	0	0	ch.8 (AN8 pin)
0	1	0	0	1	ch.9 (AN9 pin)
0	1	0	1	0	ch.10 (AN10 pin)
0	1	0	1	1	ch.11 (AN11 pin)
0	1	1	0	0	ch.12 (AN12 pin)
0	1	1	0	1	ch.13 (AN13 pin)
0	1	1	1	0	ch.14 (AN14 pin)
0	1	1	1	1	ch.15 (AN15 pin)
1	0	0	0	0	ch.16 (AN16 pin)
1	0	0	0	1	ch.17 (AN17 pin)
1	0	0	1	0	ch.18 (AN18 pin)
1	0	0	1	1	ch.19 (AN19 pin)
1	0	1	0	0	ch.20 (AN20 pin)
1	0	1	0	1	ch.21 (AN21 pin)
1	0	1	1	0	ch.22 (AN22 pin)
1	0	1	1	1	ch.23 (AN23 pin)
1	1	0	0	0	Setting prohibited
1	1	0	0	1	
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	

CCH4	CCH3	CCH2	CCH1	CCH0	Explanation
1	1	1	1	0	Setting prohibited
1	1	1	1	1	

<Note>

If the CMD0 bit is set to "1" to compare the conversion results of all channels, the settings of these bits are ignored.

26.4.13 Sampling Time Setting Registers (ADST00, ADST10)

These registers specify the sampling time, that is, the period from the start of A/D conversion until the beginning of input voltage sampling when the sampled voltage is held in the sample-and-hold circuit. The A/D conversion time is the total of the sampling time and compare time.
2 ADST registers are provided to set the sampling time. After the sampling time is set in each register, the sampling time select registers (ADSS20 to ADSS00) can be used to specify the register that has the set sampling time to be used for each channel.

Figure 26.4-16 shows the bit configuration of the sampling time setting registers (ADST00, ADST10).

Figure 26.4-16 Bit configuration of the sampling time setting registers (ADST00, ADST10)

Sampling time setting registers 00 (ADST00)

bit	15	14	13	12	11	10	9	8
	STX01	STX00	ST05	ST04	ST03	ST02	ST01	ST00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

Sampling time setting registers 10 (ADST10)

bit	7	6	5	4	3	2	1	0
	STX11	STX10	ST15	ST14	ST13	ST12	ST11	ST10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	0	0	0	0	0

R/W: Read/Write

<Notes>

- Write to these registers while A/D conversion is stopped.
- For details of the sampling time, see "■ A/D conversion time" in "26.6 Explanation of Operations and Setting Procedure Examples".

● Sampling time setting registers 00 (ADST00)

These registers set the first sampling time.

[bit15, bit14]: STX01, STX00 (Sampling time Nx setting bit)

These bits set a value (N) by which a value set in the ST05 to ST00 bits is multiplied.

STX01	STX00	Explanation
0	0	Multiplies the value by 1.
0	1	Multiplies the value by 4.
1	0	Multiplies the value by 8.
1	1	Multiplies the value by 16.

[bit13 to bit8]: ST05 to ST00 (Sampling time setting bit)

These bits set the value used to determine the sampling time.

A value that is set in these bits is used to determine the sampling time based on the following formula:

Sampling time = peripheral clock (PCLK) period $\times$ (ST + 1) $\times$ STX

ST: Value set in the ST05 to ST00 bits

STX: N (multiplier) set in the STX01 and STX00 bits

Example: ST05 to ST00 = 9, STX01, STX00 = 01 (multiply by 4), peripheral clock (PCLK) = 20 MHz (50 ns)

Sampling time = 50 ns $\times$ (9 + 1) $\times$ 4 = 2 μs

<Notes>

- If "00" (multiply the setting value by 1) is set in the STX01 and STX00 bits, set "3" or a higher number in the ST05 to ST00 bits.
- For details of the sampling time, see "■ A/D conversion time" in "26.6 Explanation of Operations and Setting Procedure Examples".
- Sampling time setting registers 00 (ADST00) must be set such that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Data Sheet".

● Sampling time setting register 10 (ADST10)

These registers set the second sampling time.

[bit7, bit6]: STX11, STX10 (Sampling time Nx setting bit)

These bits set a value (N) by which a value set in the ST15 to ST10 bits is multiplied.

STX11	STX10	Explanation
0	0	Multiplies the value by 1.
0	1	Multiplies the value by 4.
1	0	Multiplies the value by 8.
1	1	Multiplies the value by 16.

[bit5 to bit0]: ST15 to ST10 (Sampling time setting bit)

These bits set the value used to determine the sampling time.

A value that is set in these bits is used to determine the sampling time based on the following formula:

Sampling time = peripheral clock (PCLK) period $\times$ (ST + 1) $\times$ STX

ST: Value set in the ST15 to ST10 bits

STX: N (multiplier) set in the STX11 and STX10 bits

Example: ST15 to ST10 = 9, STX11, STX10 = 01 (multiply by 4), peripheral clock (PCLK) = 20 MHz (50 ns)

Sampling time = 50 ns $\times$ (9 + 1) $\times$ 4 = 2 μs

<Notes>

- If "00" (multiply the setting value by 1) is set in the STX11 and STX10 bits, set "3" or a higher number in the ST15 to ST10 bits.
- For details of the sampling time, see "■ A/D conversion time" in "26.6 Explanation of Operations and Setting Procedure Examples".
- Sampling time setting registers 10 (ADST10) must be set such that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Data Sheet".

26.4.14 Sampling Time Select Registers (ADSS20 to ADSS00)

These registers are used to select the A/D sampling time.
The sampling time to be used for each channel can be selected from that set in sampling time setting registers 00 (ADST00) or that set in sampling time setting registers 10 (ADST10).

Figure 26.4-17 shows the bit configuration of the sampling time select registers (ADSS20 to ADSS00).

Figure 26.4-17 Bit configuration of the sampling time select registers (ADSS20 to ADSS00)

Sampling time select registers 20 (ADSS20)

bit	7	6	5	4	3	2	1	0
	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Sampling time select registers 10 (ADSS10)

bit	7	6	5	4	3	2	1	0
	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Sampling time select registers 00 (ADSS00)

bit	7	6	5	4	3	2	1	0
	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Write to these registers while A/D conversion is stopped.

ADSS20 to ADSS00: TS23 to TS0 (Sampling time selection bits)

These bits specify for each channel whether to use the sampling time that is set in one of sampling time setting registers 00 (ADST00) or sampling time setting registers 10 (ADST10).

Written Value	Explanation
0	Use a sampling time that is set in sampling time setting registers 00 (ADST00).
1	Use a sampling time that is set in sampling time setting registers 10 (ADST10).

The TS23 bit corresponds to ch.23 (AN23 pin), the TS22 bit corresponds to ch.22 (AN22 pin), ... the TS1 bit corresponds to ch.1 (AN1 pin), and the TS0 bit corresponds to ch.0 (AN0 pin).

26.4.15 Compare Time Setting Registers (ADCT0)

These registers set the compare time in the A/D conversion time. The A/D conversion time is the total of the sampling time and compare time.

Figure 26.4-18 shows the bit configuration of the compare time setting registers (ADCT0).

Figure 26.4-18 Bit configuration of the compare time setting registers (ADCT0)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	CT2	CT1	CT0
Attribute	-	-	-	-	-	R/W	R/W	R/W
Initial value	X	X	X	X	X	1	1	1

R/W: Read/Write

-: Undefined

X: Undefined

<Note>

Write to this register while A/D conversion is stopped.

[bit7 to bit3]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit2 to bit0]: CT2 to CT0 (Compare time setting bits)

These bits set the value used to determine the compare time.

A value that is set in these bits is used to determine the compare time based on the following formula:

Compare time = { (CT + 1) × 10 + 4} × peripheral clock (PCLK) period

CT: Value set in these bits

Example: CT = 1, peripheral clock (PCLK) = 20 MHz (50 ns)

Compare time = {(1 + 1) × 10 + 4} × 50 ns = 1.2 μs

<Note>

For details of the compare time, see "■ A/D conversion time" in "26.6 Explanation of Operations and Setting Procedure Examples",

---

26.5 Interrupts

An interrupt request can be generated in the following case(s):

- Data has been stored in the predetermined number of stages in the FIFO during A/D scan conversion. (Scan conversion interrupt request)
- Data has been stored in the predetermined number of stages in the FIFO during A/D priority conversion. (Priority conversion interrupt request)
- An attempt has been made to save the next conversion result to a full FIFO. (FIFO overrun interrupt request)
- The conversion result satisfies the interrupt request generation conditions when the comparison function is used. (Conversion result comparison interrupt request)

■ A/D scan conversion interrupt request

Table 26.5-1 outlines the interrupt requests of A/D scan conversion.

Table 26.5-1 Interrupt requests of A/D scan conversion

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Clearing of Interrupt Request
Scan conversion interrupt request	SCIF bit = 1 in an ADCR	SCIE bit = 1 in an ADCR	Write "0" to the SCIF bit in the ADCR.
FIFO overrun interrupt request	SOVR bit = 1 in an SCCR	OVRIE bit = 1 in an ADCR	Write "0" to the SOVR bit in the SCCR.
Conversion result comparison interrupt request	CMPIF bit = 1 in an ADCR	CMPIE bit = 1 in an ADCR	Write "0" to the CMPIF bit in the ADCR.

ADCR: A/DC control register (ADCR0)

SCCR: Scan conversion control register (SCCR0)

■ A/D priority conversion interrupt request

Table 26.5-2 outlines the interrupt requests of A/D priority conversion.

Table 26.5-2 Interrupt requests of A/D priority conversion

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Clearing of Interrupt Request
Priority conversion interrupt request	PCIF bit = 1 in an ADCR	PCIE bit = 1 in an ADCR	Write "0" to the PCIF bit in the ADCR.
FIFO overrun interrupt request	POVR bit = 1 in a PCCR	OVRIE bit = 1 in an ADCR	Write "0" to the POVR bit in the PCCR.
Conversion result comparison interrupt request	CMPIF bit = 1 in an ADCR	CMPIE bit = 1 in an ADCR	Write "0" to the CMPIF bit in the ADCR.

ADCR: A/DC control register (ADCR0)

PCCR: Priority conversion control register (PCCR0)

<Notes>

- If generation of interrupt requests is enabled while the interrupt request flag is "1", an interrupt request is generated at the same time.

  Execute any of the following processing when enabling the generation of the interrupt requests:

  - Clear interrupt requests before enabling the generation of interrupt requests.
  - Clear interrupt requests simultaneously with interrupts enabled.
- For the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For details of the setting of interrupt levels, see "CHAPTER 10 Interrupt Controller".

■ Activating DMA transfer upon an interrupt

DMA transfer can be activated when one of the following interrupt requests is generated:

- Scan conversion interrupt request
- Priority conversion interrupt request

For details of DMA transfer, see "26.6.4 Activating the DMA Controller (DMAC)".

26.6 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the 10-bit A/D converter. Also, examples of procedures for setting the operating state are shown.

■ Overview

The 10-bit A/D converter enables A/D conversion by allowing analog signal input from the pin corresponding to each bit in the A/D channel enable register (ADCHE).

For details of the A/D channel enable register (ADCHE), see "15.4.6 A/D Channel Enable Register (ADCHE)" in "CHAPTER 15 I/O Ports".

The 10-bit A/D converter performs the following two types of conversion:

- A/D scan conversion

  Any selected channel is converted.

  Two conversion modes are available. One is single conversion mode in which the signals from the selected channel are converted only once, and the other is repeat conversion mode in which the signals from the selected channel are converted repeatedly.

- A/D priority conversion

  High-priority A/D conversion is performed soon after an activation trigger for the conversion is generated, because the trigger stops A/D scan conversion. The two priority levels are priority 1 and priority 2. Priority 1 > priority 2.

Table 26.6-1 summarizes the differences between A/D scan conversion and A/D priority conversion.

Table 26.6-1 Differences between A/D scan conversion and A/D priority conversion

	A/D Scan Conversion	A/D Priority Conversion	
		Priority 1	Priority 2
Supported channels	Up to 24 channels are selected arbitrarily from all 24 channels.	1 channel is selected from ch.0 to ch.7	1 channel is selected from the 24 channels.
Conversion activation trigger	Software Detection of a rising edge of the TOUT signal of base timer ch.0	Detection of a falling edge at the ADTRG0 pin	Software Detection of a rising edge of the TOUT signal of base timer ch.2
Restart	Enabled	Disabled	
FIFO	16 levels	4 levels	

■ Priority and state transition

Table 26.6-2 lists A/D conversion priorities.

Table 26.6-2 A/D conversion priority

Priority	A/D Conversion Type
1	A/D priority conversion of priority 1
2	A/D priority conversion of priority 2
3	A/D scan conversion

If A/D conversion of a different priority is activated while A/D conversion is already in progress, operations are performed as follows:

- **The A/D conversion activated while A/D conversion is already in progress has a higher priority.**

  The A/D conversion in progress is stopped and the A/D conversion of the higher priority is executed.

  After the higher-priority conversion is completed, the stopped A/D conversion is restarted.

  Example: An A/D priority conversion activation trigger is generated during A/D scan conversion

  The A/D scan conversion is interrupted, and A/D priority conversion begins. After the A/D priority conversion is completed, the A/D scan conversion resumes from the channel at which it interrupted.

  Example: An activation trigger for A/D priority conversion of priority 1 is generated during A/D priority conversion of priority 2.

  The A/D priority conversion of priority 2 is interrupted, and the A/D priority conversion of priority 1 begins. After the A/D priority conversion of priority 1 is completed, the A/D priority conversion of priority 2 resumes.

- **The A/D conversion activated while A/D conversion is already in progress has a lower priority.**

  The activation trigger for the A/D conversion of a lower priority is held, and the A/D conversion in progress is executed continuously.

  After the A/D conversion in progress is completed, the A/D conversion whose activation trigger has been held begins automatically.

  Example: An activation trigger for A/D priority conversion of priority 2 is generated during A/D priority conversion of priority 1.

  The activation trigger for the A/D priority conversion of priority 2 is held, and the A/D priority conversion of priority 1 is executed continuously.

  After the A/D priority conversion of priority 1 is completed, the A/D priority conversion of priority 2 begins automatically.

  Example: An activation trigger for A/D scan conversion is generated during A/D priority conversion of priority 1.

  The activation trigger for the A/D scan conversion is held, and the A/D priority conversion of priority 1 is executed continuously.

  After the A/D priority conversion of priority 1 is completed, the A/D scan conversion begins automatically.

  Example: An activation trigger for A/D scan conversion is generated during A/D priority conversion of priority 2.

  The activation trigger for the A/D scan conversion is held, and the A/D priority conversion of priority 2 is executed continuously.

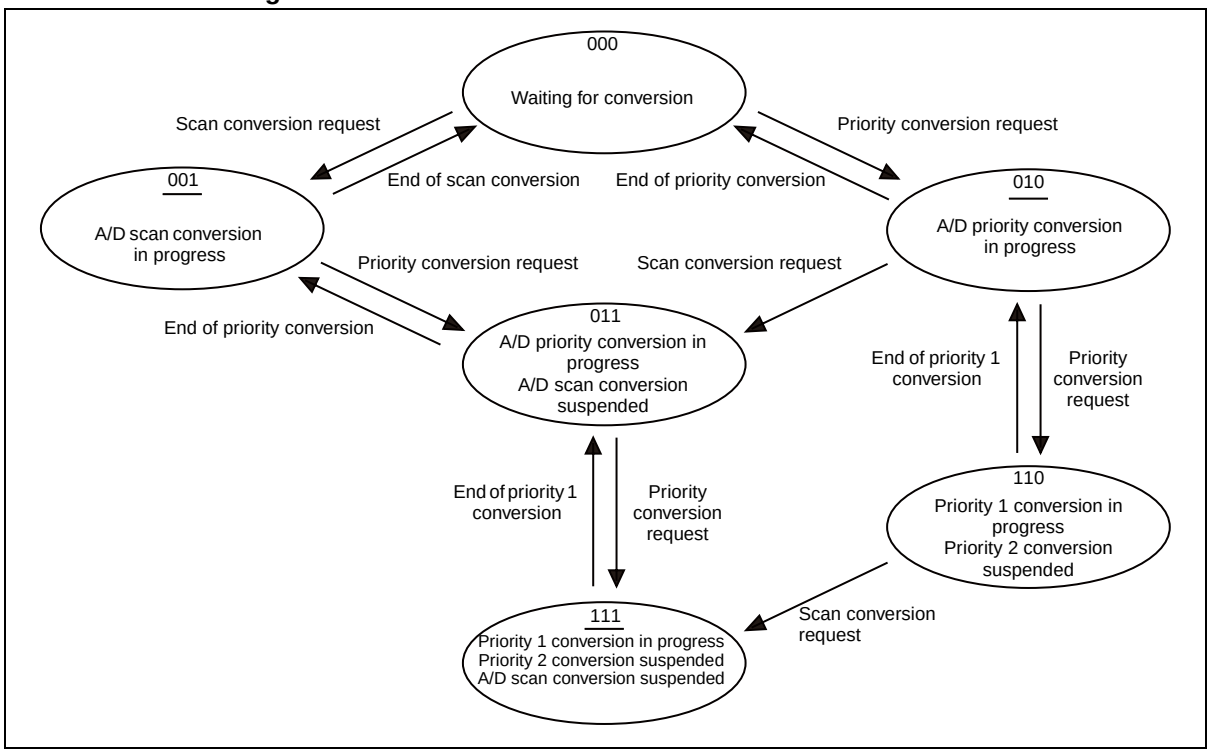
  After the A/D priority conversion of priority 2 is completed, the A/D scan conversion begins automatically.

- **The A/D conversion activated during A/D priority conversion has the same priority.**

  An activation trigger with the same priority is ignored. (The ignored activation trigger will not be reactivated.)

Figure 26.6-1 show state transitions of the 10-bit A/D converter.

Figure 26.6-1 State transitions of the 10-bit A/D converter



As shown in Figure 26.6-1, the states of the 10-bit A/D converter can be checked with the PCNS, PCS, and SCS bits in the A/DC status registers (ADSR0).

Table 26.6-3 shows the relationship between bits and operating states.

Table 26.6-3 Relationship between bits and operating states

PCNS	PCS	SCS	Explanation
0	0	0	Waiting for conversion
0	0	1	A/D scan conversion in progress
0	1	0	A/D priority conversion in progress
0	1	1	A/D priority conversion in progress, with A/D scan conversion suspended
1	1	0	Priority 1 A/D priority conversion in progress, with priority 2 conversion suspended
1	1	1	Priority 1 A/D priority conversion in progress, with priority 2 conversion and scan conversion suspended

■ Operation using the A/D comparison function

The A/D comparison function compares the eight high-order bits (bit9 to bit2) of A/D conversion results with a preset value in the A/D comparison data setting registers (CMPD0). If the comparison result satisfies the requirements set in an A/D comparison control register (CMPCR0), the function generates a conversion result comparison interrupt request.

The CMPEN bit must be set to "1" in the A/D comparison control register (CMPCR0) to enable the comparison function before conversion is started.

The comparison function can be used even with a full FIFO because a comparison is made before A/D conversion results are stored in the FIFO.

For details of the comparison function, see "26.4.11 A/D Comparison Data Setting Registers (CMPD0)", and "26.4.12 A/D Comparison Control Registers (CMPCR0)".

■ A/D conversion time

The A/D conversion time is the total of the sampling time and compare time.

To determine the A/D conversion time, add the sampling time and compare time.

● Sampling time

The sampling time can be set in each of the sampling time setting registers (ADST00, ADST10).

The sampling time select register (ADSS20 to ADSS00) can be used to specify the register that has the set sampling time to be used for each channel. Therefore, the sampling time can be set individually for channels with different external impedances.

The formula for calculating the sampling time is as follows:

Sampling time = peripheral clock (PCLK) period × (ST + 1) × STX

ST: Value that is set in the ST05 to ST00/ST15 to ST10 bits in a sampling time setting register (ADST00, ADST10)

STX: Multiplier that is set in the STX01, STX00/STX11, and STX10 bits in a sampling time setting register (ADST00, ADST10)

<Notes>

- If "00" (multiply the setting value by 1) is set in the STX01 and STX00 bits, set "3" or a higher number in the ST05 to ST00/ST15 to ST10 bits.
- Sampling time setting registers 00 (ADST00) must be set so that the sampling time in the electrical characteristics is satisfied. For details of the electrical characteristics, see "Data Sheet".

Table 26.6-4 and Table 26.6-5 show sampling time setting examples.

Table 26.6-4 Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 00) (1 / 2)

Register value (N) STx5 to STx0	Sampling Time [µs]			Maximum External Impedance [kΩ]		
	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
0	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
1	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
2	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
3	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
4	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
5	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
6	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
7	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
8	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
9	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
10	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
11	0.400	Setting prohibited	Setting prohibited	1.400	-	-
12	0.433	0.406	Setting prohibited	1.400	1.400	-
13	0.467	0.438	0.424	1.563	1.400	1.400
14	0.500	0.469	0.455	2.053	1.593	1.400
15	0.533	0.500	0.485	2.543	2.053	1.830
16	0.567	0.531	0.515	3.033	2.513	2.276
17	0.600	0.563	0.545	3.524	2.972	2.721
18	0.633	0.594	0.576	4.014	3.432	3.167
19	0.667	0.625	0.606	4.504	3.891	3.613
20	0.700	0.656	0.636	4.994	4.351	4.058

Table 26.6-4 Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 00) (2 / 2)

Register value (N) STx5 to STx0	Sampling Time [µs]			Maximum External Impedance [kΩ]		
	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
...	...	...	...	...	...	...
36	1.233	1.156	1.121	12.837	11.704	11.188
37	1.267	1.188	1.152	13.327	12.163	11.634
38	1.300	1.219	1.182	13.818	12.623	12.080
...	...	...	...	...	...	...
42	1.433	1.344	1.303	15.778	14.461	13.862
43	1.467	1.375	1.333	16.269	14.921	14.308
...	...	...	...	...	...	...
52	1.767	1.656	1.606	20.680	19.057	18.319
53	1.800	1.688	1.636	21.171	19.516	18.764
...	...	...	...	...	...	...
62	2.100	1.969	1.909	25.582	23.652	22.775
63	2.133	2.000	1.939	26.073	24.112	23.220

PCLK : peripheral clock (PCLK) frequency

Table 26.6-5 Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 10) (1 / 2)

Register Value (N) STx5 to STx0	Sampling Time [µs]			Maximum External Impedance [kΩ]		
	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
0	Setting prohibited	Setting prohibited	Setting prohibited	-	-	-
1	0.533	0.500	0.485	2.543	2.053	1.830
2	0.800	0.750	0.727	6.465	5.729	5.395
3	1.067	1.000	0.970	10.386	9.406	8.960
4	1.333	1.250	1.212	14.308	13.082	12.525
5	1.600	1.500	1.455	18.229	16.759	16.090
6	1.867	1.750	1.697	22.151	20.435	19.655
7	2.133	2.000	1.939	26.073	24.112	23.220
8	2.400	2.250	2.182	29.994	27.788	26.786
9	2.667	2.500	2.424	33.916	31.465	30.351
10	2.933	2.750	2.667	37.837	35.141	33.916
11	3.200	3.000	2.909	41.759	38.818	37.481
12	3.467	3.250	3.152	45.680	42.494	41.046
13	3.733	3.500	3.394	49.602	46.171	44.611
14	4.000	3.750	3.636	53.524	49.847	48.176
15	4.267	4.000	3.879	57.445	53.524	51.741
16	4.533	4.250	4.121	61.367	57.200	55.306
17	4.800	4.500	4.364	65.288	60.876	58.871
18	5.067	4.750	4.606	69.210	64.553	62.436
19	5.333	5.000	4.848	73.131	68.229	66.001
20	5.600	5.250	5.091	77.053	71.906	69.566
...	...	...	...	...	...	...
36	9.867	9.250	8.970	139.798	130.729	126.607
37	10.133	9.500	9.212	143.720	134.406	130.172
38	10.400	9.750	9.455	147.641	138.082	133.737
...	...	...	...	...	...	...
42	11.467	10.750	10.424	163.327	152.788	147.998
43	11.733	11.000	10.667	167.249	156.465	151.563
...	...	...	...	...	...	...

Table 26.6-5 Sampling time setting examples (for STX01, STX00/STX11, STX10 bits = 10) (2 / 2)

Register Value (N) STx5 to STx0	Sampling Time [µs]			Maximum External Impedance [kΩ]		
	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz	PCLK= 30MHz	PCLK= 32MHz	PCLK= 33MHz
52	14.133	13.250	12.848	202.543	189.553	183.648
53	14.400	13.500	13.091	206.465	193.229	187.213
...	...	...	...	...	...	...
62	16.800	15.750	15.273	241.759	226.318	219.299
63	17.067	16.000	15.515	245.680	229.994	222.864

PCLK : peripheral clock (PCLK) frequency

● Compare time

The compare time setting registers (ADCT0) specify the compare time.
The formula for calculating the compare time is as follows:

Compare time = { (CT + 1) × 10 + 4} × peripheral clock (PCLK) period

CT: Value that is set in the CT2 to CT0 bits in a compare time setting register (ADCT0)

Table 26.6-6 shows a compare time setting example.

Table 26.6-6 Compare time setting example

Register Value (N) CT2 to CT0	Compare Time		
	PCLK = 30 MHz	PCLK = 32 MHz	PCLK = 33 MHz
0	Setting prohibited	Setting prohibited	Setting prohibited
1	0.80 µs	0.75 µs	0.73 µs
2	1.13 µs	1.06 µs	1.03 µs
3	1.47 µs	1.38 µs	1.33 µs
4	1.80 µs	1.69 µs	1.64 µs
5	2.13 µs	2.00 µs	1.94 µs
6	2.47 µs	2.31 µs	2.24 µs
7 (initial value)	2.80 µs	2.63 µs	2.55 µs

PCLK peripheral clock (PCLK) frequency

* This table covers only compare time data.

26.6.1 Operation of A/D Scan Conversion

Channels are selected by the scan conversion input select registers (SCIS20 to SCIS00) sequentially.

■ Overview

A/D scan conversion is performed in one of the following two conversion modes:

- Single conversion mode

  The channel specified by a scan conversion input select register (SCIS20 to SCIS00) is converted only once.

- Repeat conversion mode

  The channel specified by a scan conversion input select register (SCIS20 to SCIS00) is converted repeatedly.

Also, the operation that is performed varies depending on whether only one channel is selected or multiple channels are selected with a scan conversion input select register (SCIS20 to SCIS00).

Table 26.6-7 shows the order of conversion in each conversion mode.

Table 26.6-7 Conversion mode and order of conversion

Conversion Mode	Selected Channel	Conversion Order
Single conversion mode (RPT bit = 0 in an SCCR)	ch.3	ch.3 → conversion stopped
	ch.3, ch.5, ch.10, ch.23	ch.3 → ch.5 → ch.10 → ch.23 → conversion stopped
Repeat conversion mode (RPT bit = 1 in an SCCR)	ch.3	ch.3 → ch.3 → ch.3 → ch.3 ↑ ↓ ch.3 ← ch.3 ← ch.3 ← ch.3
	ch.3, ch.5, ch.10, ch.23	ch.3 → ch.5 → ch.10 → ch.23 ↑ ↓ ch.23 ← ch.10 ← ch.5 ← ch.3

SCCR: scan conversion control register (SCCR0)

<Note>

The 10-bit A/D converter enables A/D conversion by allowing analog signal input with the A/D channel enable register (ADCHE).

For details of the A/D channel enable register (ADCHE), see "15.4.6 A/D Channel Enable Register (ADCHE)" in "CHAPTER 15 I/O Ports".

■ Operation in single conversion mode

Writing "0" to the RPT bit in a scan conversion control register (SCCR0) sets single conversion mode.

In this mode, the channel specified by a scan conversion input select register (SCIS20 to SCIS00) is converted only once.

● Activation

The channel to be converted is selected with the scan conversion input select register (SCIS20 to SCIS00) in one of the following ways, and the 10-bit A/D converter is then activated:

- Write "1" to the SSTR bit in a scan conversion control register (SCCR0).
- Set the SHEN bit in a scan conversion control register (SCCR0) to enable timer activation (SHEN = 1), and input the rising edge of a TOUT signal of base timer ch.0.

If either of the above activation operations is performed during A/D scan conversion, the A/D scan conversion is immediately stopped/initialized, and the A/D scan conversion resumes later (reactivated).

● Single-channel conversion

Only one channel to be converted is selected with a scan conversion input select register (SCIS20 to SCIS00).

When started, the 10-bit A/D converter activates the conversion operation for the selected channel, and the SCS bit in an A/DC status register (ADSR0) changes to "1".

After the conversion of the selected channel is completed, the conversion result and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the conversion operation is then stopped. Then, the SCS bit in the A/DC status register (ADSR0) is cleared to "0".

The conversion results stored in the FIFO can be read from a scan conversion FIFO data register (SCFD0).

● Multichannel conversion

Two or more channels to be converted are selected with a scan conversion input select register (SCIS20 to SCIS00).

When started, the 10-bit A/D converter begins converting the selected channels in ascending order of channel number. The SCS bit in an A/DC status register (ADSR0) then changes to "1".

After the conversion of one channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the converter then begins converting the next channel.

The channels not selected by the scan conversion input select register (SCIS20 to SCIS00) remain unconverted.

In the A/D scan conversion FIFO, the number of stages storing conversion results and information on each converted channel is changed every time the channel subject to conversion changes.

The 10-bit A/D converter stops operating after converting on all the channels selected by the scan conversion input select register (SCIS20 to SCIS00). Then, the SCS bit in the A/DC status register (ADSR0) is cleared to "0".

The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading, see "■ Operation of A/D scan conversion" in "26.6.3 FIFO Operations".

■ Operation in repeat conversion mode

Writing "1" to the RPT bit in a scan conversion control register (SCCR0) enables single conversion mode.

In this mode, the channel specified by a scan conversion input select register (SCIS20 to SCIS00) is converted repeatedly.

As in single conversion mode, to activate the 10-bit A/D converter, select a channel.

● Single-channel conversion

Only one channel to be converted is selected with a scan conversion input select register (SCIS20 to SCIS00).

When started, the 10-bit A/D converter activates the conversion operation for the selected channel, and the SCS bit in an A/DC status register (ADSR0) changes to "1".

After the conversion of the selected channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and conversion of the same channel is then repeated.

To stop conversion, write "0" to a scan conversion control register (SCCR0).

The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading conversion results, see "■ Operation of A/D scan conversion" in "26.6.3 FIFO Operations".

● Multichannel conversion

Two or more channels to be converted are selected with a scan conversion input select register (SCIS20 to SCIS00).

When activated, the 10-bit A/D converter begins converting on the selected channels sequentially in ascending order of channel number. The SCS bit in an A/DC status register (ADSR0) then changes to "1".

After the conversion of one channel is completed, the conversion results and information on the converted channel are stored in the first level of the A/D scan conversion FIFO, and the converter then begins converting the next channel.

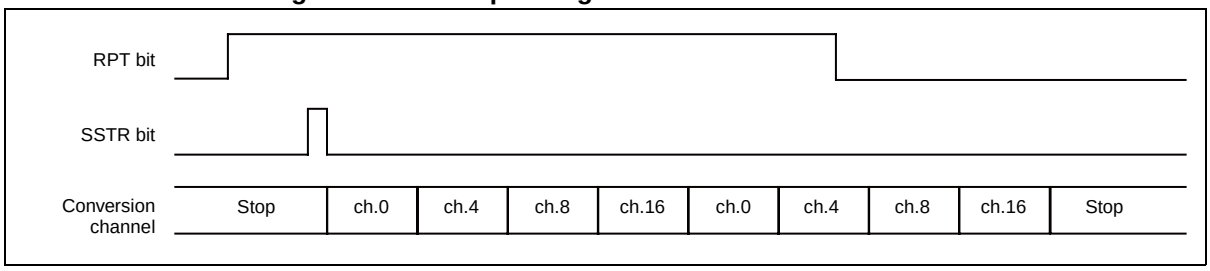
The channels not selected by the scan conversion input select register (SCIS20 to SCIS00) remain unconverted.

After conversion of all the selected channels is completed, the second round of conversion begins for the channels sequentially in ascending order of channel number.

To stop conversion, write "0" in the RPT bit of the scan conversion control register (SCCR0). Conversion stops when all channels selected by the scan conversion input select register (SCIS20 to SCIS00) have been converted.

Figure 26.6-2 shows the stop timing in multichannel conversion.

Figure 26.6-2 Stop timing in multichannel conversion



The conversion results stored in the FIFO can be read sequentially from a scan conversion FIFO data register (SCFD0). For details of reading, see "■ Operation of A/D scan conversion" in "26.6.3 FIFO Operations".

26.6.2 Operation of A/D Priority Conversion

High-priority A/D conversion is performed soon after an activation trigger for the conversion is generated, because the trigger stops A/D scan conversion. There are two priority levels.

■ Overview

Two priority levels can be selected depending on the activation trigger. Priority 1 has a higher priority than priority 2.

The channels that can be set vary depending on the priority.

Table 26.6-8 shows the relationship among priorities, channels, and activation triggers.

Table 26.6-8 Relationship among, priorities, channels, and activation triggers

	Priority 1	Priority 2
Priority	1	2
Supported channel	1 channel is selected from ch.0 to ch.7	1 channel is selected from the 24 channels.
Activation trigger	Detection of a falling edge at the ADTRG0 pin	Software Detection of a rising edge of the TOUT signal of base timer ch.2

<Notes>

- The 10-bit A/D converter enables A/D conversion by allowing analog signal input with the A/D channel enable register (ADCHE).

  For details of the A/D channel enable register (ADCHE), see "15.4.6 A/D Channel Enable Register (ADCHE)" in "CHAPTER 15 I/O Ports".
- A/D conversion can be reactivated regardless of priority after A/D priority conversion.
- Only one channel can be converted in A/D priority conversion.

■ Conversion with priority 1

This conversion has the highest priority. When an activation trigger with priority 1 is generated, any A/D scan conversion or A/D priority conversion that is in progress is immediately stopped, and the conversion with priority 1 begins.

● Selecting a channel

Only one channel is selected from ch.0 to ch.7 and set in the P1A2 to P1A0 bits in a priority conversion input select register (PCIS0).

● Conversion

An activation trigger for A/D priority conversion of priority 1 is generated when a falling edge is detected at the ADTRG0 pin after the PEEN bit in a priority conversion control register (PCCR0) is set to "1" to enable external activation.

If A/D scan conversion or A/D priority conversion of priority 2 is in progress, it is immediately interrupted, and conversion of the specified channel with priority 1 begins. The PCS bit in an A/DC status register (ADSR0) then changes to "1".

After the conversion is completed, the conversion results and information on the channel subject to conversion are stored in the FIFO for A/D priority conversion, and the PCS bit in the A/DC status register (ADSR0) is cleared to "0". The interrupted conversion is then restarted.

The A/D priority conversion results stored in the FIFO can be read from a priority conversion FIFO data register (PCFD0). For details of reading, see "■ Operation of A/D priority conversion" in "26.6.3 FIFO Operations".

For details of the operation performed when an activation trigger with a different priority is generated while A/D priority conversion of priority 1 is in progress, see "■ Priority and state transition in "26.6 Explanation of Operations and Setting Procedure Examples".

<Note>

If an activation trigger for A/D conversion of the same level (priority 1) is generated while A/D priority conversion of priority 1 is in progress, the conversion in progress is continued and the new activation trigger is ignored.

■ Conversion with priority 2

This conversion has the second highest priority. When an activation trigger of priority 2 is generated, any A/D scan conversion in progress is immediately stopped, and the conversion with priority 2 begins.

● Selecting a channel

Only one channel to be converted is selected from all 24 channels and set in the P2A4 to P2A0 bits in a priority conversion input select register (PCIS0).

● Conversion

An activation trigger with priority 2 is generated in one of the following ways:

- Write "1" to the PSTR bit in a priority conversion control register (PCCR0).
- A rising edge of the TOUT signal of base timer ch.2 is detected after the PHEN bit in the priority conversion control register (PCCR0) is set to "1" to enable timer activation.

When an activation trigger is generated, A/D priority conversion of priority 2 is activated and the PCS bit in an A/DC status register (ADSR0) changes to "1" as follows:

- If the 10-bit A/D converter is not activated: The 10-bit A/D converter is activated to start conversion of the specified channel with priority 2.
- If A/D scan conversion is in progress: The A/D scan conversion in progress is immediately interrupted, and conversion of the specified channel with priority 2 begins.
- If A/D priority conversion of priority 1 is in progress: The activation trigger with priority 2 is held, and A/D priority conversion of priority 2 is started after A/D priority conversion of priority 1 is completed.

After A/D priority conversion of priority 2 is completed, the conversion results and information on the channel subject to conversion are stored in the FIFO for A/D priority conversion, and the PCS bit in an A/DC status register (ADSR0) is cleared to "0". The interrupted conversion is then restarted.

The A/D priority conversion results stored in the FIFO can be read from a priority conversion FIFO data register (PCFD0). For details of reading, see "■ Operation of A/D priority conversion" in "26.6.3 FIFO Operations".

For details of the operation performed when an activation trigger with a different priority is generated while A/D priority conversion of priority 2 is in progress, see "■Priority and state transition in "26.6 Explanation of Operations and Setting Procedure Examples".

<Note>

No conversion operation can be reactivated during A/D priority conversion. If an activation trigger for A/D conversion of the same level (priority 2) is generated while A/D priority conversion of priority 2 is in progress, the conversion in progress is continued and the new activation trigger is ignored.

Example: The rising edge of a TOUT signal of base timer ch.2 may be detected after A/D priority conversion of priority 2 is activated by software. Even in this event, the conversion operation in progress is continued.

26.6.3 FIFO Operations

The 10-bit A/D converter provides 16 FIFO stages for A/D scan conversion and 4 FIFO levels for A/D priority conversion. A scan conversion interrupt request/priority conversion interrupt request can be generated when the number of stages storing the respective data reaches the predetermined number of FIFO stages.
This section explains FIFO operations and generation of interrupt requests.

■ Operation of A/D scan conversion

● Operation during A/D conversion

The SEMP bit in a scan conversion control register (SCCR0) is "1", because the FIFO for A/D scan conversion contains no data (empty) after a reset is released.

The SEMP bit changes to "0" when A/D scan conversion begins, and conversion results for 1 channel are stored in the first FIFO stage.

After conversion of the next data is completed, the conversion results are stored in the second FIFO stage. Every time that conversion of 1 channel is completed after that, the conversion results are stored in the subsequent FIFO stage.

After conversion results are written to all 16 FIFO levels, the A/D scan conversion FIFO becomes full and the SFUL bit in the scan conversion control register (SCCR0) changes to "1".

If A/D scan conversion is performed again in this state, an overrun occurs and the SOVR bit in the scan conversion control register (SCCR0) changes to "1". In this event, the conversion results are not stored in the FIFO but abandoned.

● Read operation

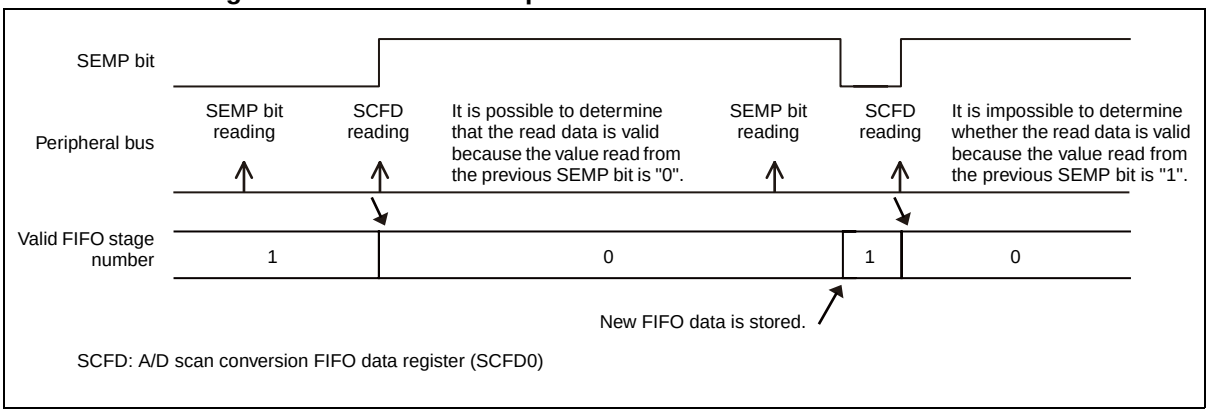
Data stored in the A/D scan conversion FIFO can be read sequentially through reading with a scan conversion FIFO data register (SCFD0).

A scan conversion FIFO data register (SCFD0) must always be read after the SEMP bit in a scan conversion control register (SCCR0) is checked to determine whether data remains in the A/D scan conversion FIFO (SEMP = 0).

If an empty A/D scan conversion FIFO is read (SEMP = 1), it is difficult to determine whether the read data is valid, and valid data may be abandoned. (This is because conversion results may be stored in a scan conversion FIFO data register (SCFD0) immediately before the FIFO is read.)

Figure 26.6-3 shows the relationship between the SEMP bit and read data.

Figure 26.6-3 Relationship between the SEMP bit and read data


SEMP bit
Peripheral bus
SEMP bit reading
SCFD reading
It is possible to determine that the read data is valid because the value read from the previous SEMP bit is "0".
SEMP bit reading
SCFD reading
It is impossible to determine whether the read data is valid because the value read from the previous SEMP bit is "1".
Valid FIFO stage number
1
0
1
0
New FIFO data is stored.
SCFD: A/D scan conversion FIFO data register (SCFD0)

<Notes>

- The registers listed below are located at adjacent addresses. If these registers are accessed at one time by word access, the registers are read regardless of the setting of the SEMP bit in a scan conversion control register (SCCR0). Do not execute word access to these registers.
  - Scan conversion control register (SCCR0)
  - Scan conversion FIFO number setting register (SFNS0)
  - Scan conversion FIFO data register (SCFD0)
- The scan conversion FIFO data registers (SCFD0) can be byte-accessed. FIFO data is shifted after the high-order byte (bit15 to bit8) is read. FIFO data will not be shifted by reading of the lower-order byte (bit7 to bit0).

● Clear operation

Writing "1" to the SFCLR bit in a scan conversion control register (SCCR0) clears the FIFO for A/D scanning conversion and changes the SEMP bit in the scan conversion control register (SCCR0) to "1".

● Scan conversion interrupt request

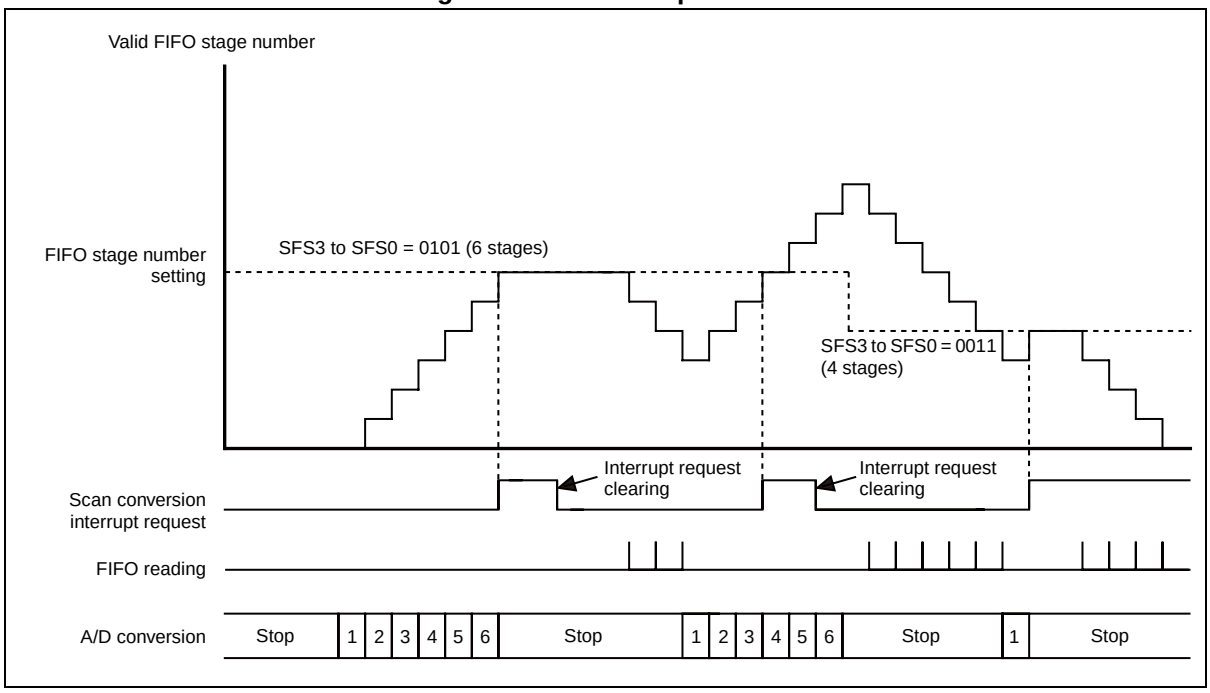
A scan conversion interrupt request can be generated when the number of stages storing conversion results reaches the specified number of FIFO stages (SCIF bit = 1 in an A/DC control register (ADCR0)).

To generate an A/D scan conversion interrupt request, perform the following processing:

- Decide the number of FIFO stages at which an interrupt request is generated, and set the number in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).
- Set the SCIE bit in an A/DC control register (ADCR0) to "1" to enable generation of scan conversion interrupt requests.

Figure 26.6-4 shows FIFO operations.

Figure 26.6-4 FIFO operations



The interrupt request generation examples shown below indicate the number of FIFO stages that is set for each conversion mode. The number of FIFO stages must be set in the SFS3 to SFS0 bits in a scan conversion FIFO number setting register (SFNS0).

- **Single channel conversion in single conversion mode**

  If the number of FIFO stages at which a scan conversion interrupt request is generated is set at 1 (SFS3 to SFS0 = 0000), a scan conversion interrupt request is generated when conversion is completed. If the number of FIFO stages is set at 2 or more (SFS3 to SFS0 = 0001 or higher), an interrupt request is not generated even after conversion of the specified channel is completed.

- **Multichannel conversion in single conversion mode**

  If the set number of FIFO stages is the same as the number of channels to be subject to conversion, a scan conversion interrupt request is generated at the end of conversion.

  Example: Generation of a scan conversion interrupt request after conversion of 3 channels

  Set 3 (SFS3 to SFS0 = 0010) for the number of FIFO stages at which a scan conversion interrupt request is generated.

  Settings can be made such that a scan conversion interrupt request is generated at a number of FIFO stages that is less than the number of the channels to be subject to conversion. In such cases, a scan conversion interrupt request can be generated at any time before the end of A/D scan conversion.

- **Single channel conversion in repeat conversion mode**

  If the number of FIFO stages at which a scan conversion interrupt request is generated is set at 1 (SFS3 to SFS0 = 0000), a scan conversion interrupt request is generated when the first round of conversion is completed.

  To generate a scan conversion interrupt request after converting on the specified channel a certain number of times, match the conversion count to the number of FIFO stages.

  Example: Generation of a scan conversion interrupt request after conversion on a single channel is performed 4 times

  Set 4 (SFS3 to SFS0 = 0011) for the number of FIFO stages at which a scan conversion interrupt request is generated.

- **Multichannel conversion in repeat conversion mode**

  The desired generation time for scan conversion interrupt requests can be selected as shown below.

  Example: Conversion of 8 channels in repeat conversion mode

  - Generate a scan conversion interrupt request after the end of the first round of conversion.

    Set 8 (SFS3 to SFS0 = 0111) for the number of FIFO stages at which a scan conversion interrupt request is generated.
  - Generate an interrupt request after the end of the second round of conversion.

    Set 16 (twice the number of channels to be subject to conversion) (SFS3 to SFS0 = 1111) for the number of FIFO stages at which a scan conversion interrupt request is generated.

<Note>

DMA transfer of the data in the FIFO can be performed when a scan conversion interrupt request is generated. For details of DMA transfer, see "26.6.4 Activating the DMA Controller (DMAC)".

● FIFO overrun interrupt request

When data has been stored in all 16 FIFO levels and the FIFO becomes full, the SFUL bit in a scan conversion control register (SCCR0) changes to "1".

Also, the OVRIE bit in an A/DC control register (ADCR0) can be set to enable generation of FIFO overrun interrupt requests (OVRIE = 1). In this state, if an attempt is made to store the next conversion result in the FIFO when the SFUL bit is "1", an overrun interrupt is generated.

<Notes>

- The data in the FIFO cannot be rewritten, even by the attempt to store the next conversion result in the full FIFO. The conversion result to be stored in this attempt is abandoned.
- The FIFO is emptied and the SEMP bit in a scan conversion control register (SCCR0) changes to "1" when the SFCLR bit in the scan conversion control register (SCCR0) is set to "1" to clear the FIFO.

■ Operation of A/D priority conversion

● Operation during A/D conversion

The PEMP bit in an A/D priority conversion control register (PCCR0) is "1", because the FIFO for A/D scan conversion contains no data (empty) after a reset is released.

The PEMP bit changes to "0" when A/D priority conversion begins, and conversion results for 1 channel are stored in the first FIFO stage.

After the next A/D priority conversion is completed, the conversion results are stored in the second FIFO level. Every time that A/D priority conversion is completed after that, the conversion results are stored in the subsequent FIFO stage.

After conversion results are written to all 4 FIFO levels, the FIFO for A/D priority conversion becomes full and the PFUL bit in the priority conversion control register (PCCR0) changes to "1".

If A/D priority conversion is performed again in this state, an overrun occurs and the POVR bit in the priority conversion control register (PCCR0) changes to "1". In this event, the conversion results are not stored in the FIFO but abandoned.

● Read operation

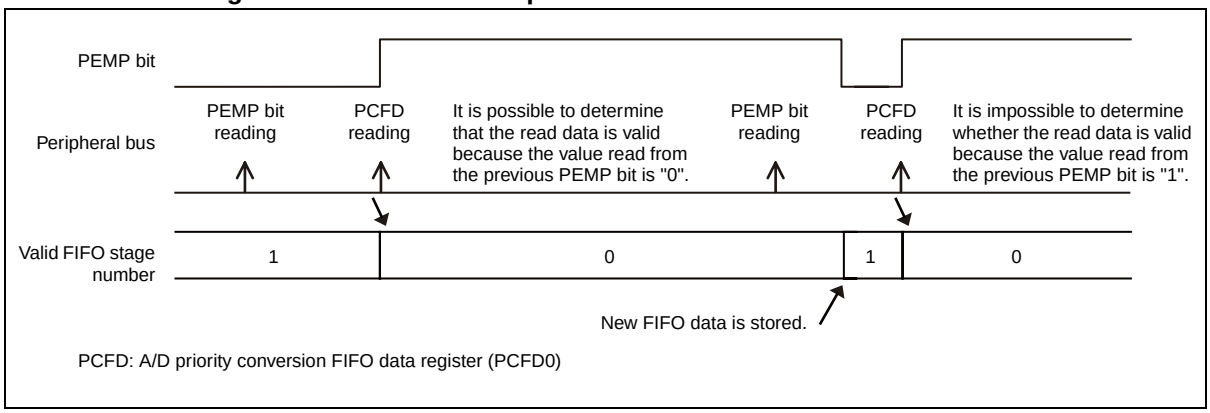
Data stored in the A/D priority conversion FIFO can be read sequentially through reading with a priority conversion FIFO data register (PCFD0).

A priority conversion FIFO data register (PCFD0) must always be read after the PEMP bit in a priority conversion control register (PCCR0) is checked to determine whether data remains in the A/D priority conversion FIFO (PEMP = 0).

If an empty A/D priority conversion FIFO is read (PEMP = 1), it is difficult to determine whether the read data is valid, and valid data may be abandoned. (This is because conversion results may be stored in a priority conversion FIFO data register (PCFD0) immediately before the FIFO is read.)

Figure 26.6-5 shows the relationship between the PEMP bit and read data.

Figure 26.6-5 Relationship between the PEMP bit and read data


PEMP bit
Peripheral bus
PEMP bit reading
PCFD reading
It is possible to determine that the read data is valid because the value read from the previous PEMP bit is "0".
PEMP bit reading
PCFD reading
It is impossible to determine whether the read data is valid because the value read from the previous PEMP bit is "1".
Valid FIFO stage number
1
0
1
0
New FIFO data is stored.
PCFD: A/D priority conversion FIFO data register (PCFD0)

<Notes>

- The registers listed below are located at adjacent addresses. If these registers are accessed at one time by word access, the registers are read regardless of the setting of the PEMP bit in a priority conversion control register (PCCR0). Do not execute word access to these registers.
  - Priority conversion control register (PCCR0)
  - Priority conversion FIFO number setting register (PFNS0)
  - Priority conversion FIFO data register (PCFD0)
- The priority conversion FIFO data registers (PCFD0) can be byte accessed. FIFO data is shifted after the high-order byte (bit15 to bit8) is read. FIFO data will not be shifted by reading of the low-order byte (bit7 to bit0).

● Clear operation

Writing "1" to the PFCLR bit in a priority conversion control register (PCCR0) clears the A/D priority conversion FIFO and changes the PEMP bit in an A/D priority conversion control register (PCCR0).

● Priority conversion interrupt request

A priority conversion interrupt request can be generated when the number of stages storing conversion results reaches the specified number of FIFO stages (PCIF bit = 1 in an A/DC control register (ADCR0)).

To generate an A/D priority conversion interrupt request, perform the following processing:

- Decide the number of FIFO stages at which an interrupt request is generated, and set the number in the PFS1 and PFS0 bits in the priority conversion FIFO number setting register (PFNS0).
- Set the PCIE bit in the A/DC control register (ADCR0) to "1" to enable generation of priority conversion interrupt requests.

If the number of FIFO stages at which a priority conversion interrupt request is generated is set to "1" (PFS1, PFS0 = 00), a priority conversion interrupt request is generated when the conversion is completed.

<Notes>

- If the number of FIFO stages at which a priority interrupt request is generated is set at 2 or more (PFS1, PFS0 = 01 or higher), no priority conversion interrupt request is generated even after A/D priority conversion is completed.
- DMA transfer of the data in the FIFO can be performed when a priority conversion interrupt request is generated. For details of DMA transfer, see "26.6.4 Activating the DMA Controller (DMAC)".

● FIFO overrun interrupt request

The PFUL bit in a priority conversion control register (PCCR0) changes to "1" when data has been stored in all of 4 FIFO levels and the FIFO becomes full.

Also, the OVRIE bit in an A/DC control register (ADCR0) can be set to enable generation of FIFO overrun interrupt requests (OVRIE = 1). In this state, if an attempt is made to store the next conversion result in FIFO when the PFUL bit is "1", an overrun interrupt is generated.

<Notes>

- The data in the FIFO cannot be rewritten, even by the attempt to store the next conversion result in the full FIFO. The conversion result to be stored in this attempt is abandoned.
- The FIFO is emptied and the PEMP bit in a priority conversion control register (PCCR0) changes to "1" when the PFCLR bit in the priority conversion control register (PCCR0) is set to "1" to clear the FIFO.

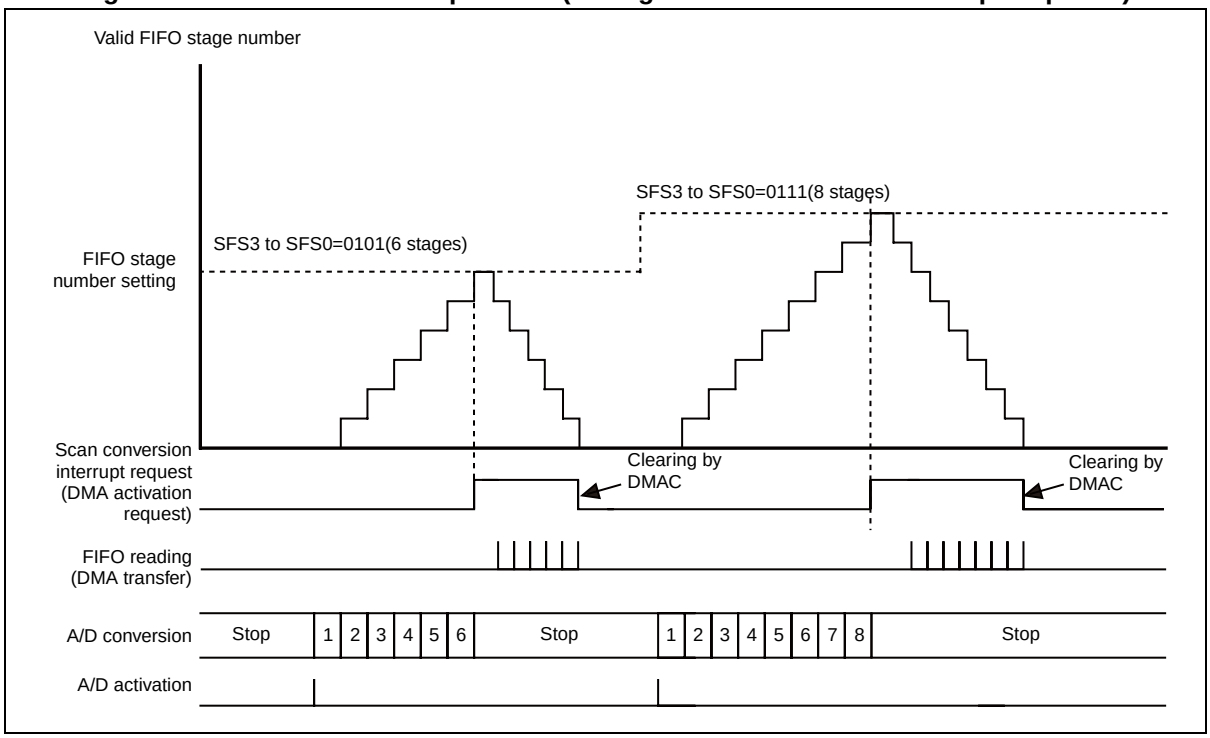
26.6.4 Activating the DMA Controller (DMAC)

DMA transfer of FIFO data is possible through scan conversion interrupt requests and priority conversion interrupt requests generated by the 10-bit A/D converter.

If the same value is set for the number of FIFO stages at which a scan conversion interrupt request/ priority conversion interrupt request is generated and the byte number for DMA transfer, DMA transfer of FIFO data can be performed in synchronization with A/D scan conversion. For details of setting the byte number for DMA transfer, see "CHAPTER 33 DMA Controller (DMAC)".

Figure 26.6-6 shows the DMA transfer operation.

Figure 26.6-6 DMA transfer operation (through scan conversion interrupt requests)



However, note that the event described below may occur while A/D conversion is repeated, such as in repeat conversion mode. In such cases, even after DMA transfer of data by the specified number of the bytes, the FIFO may still store more data corresponding to stages exceeding the number of stages at which a scan conversion interrupt request/priority conversion interrupt request is generated.

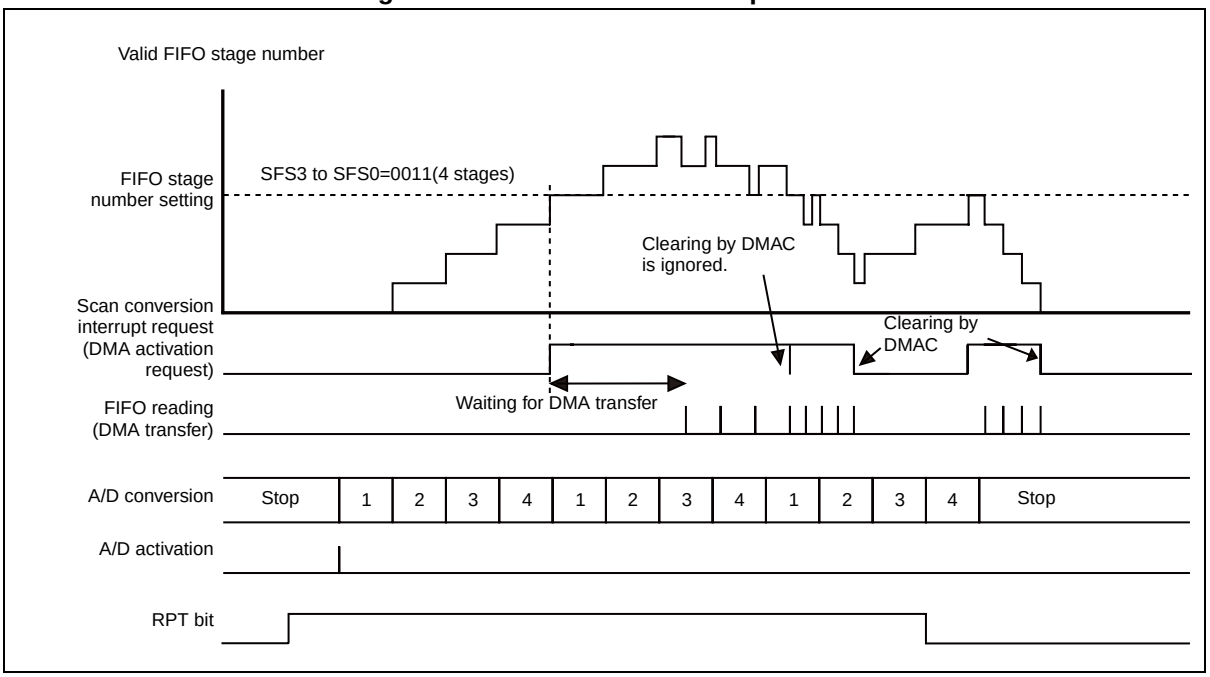
- A/D conversion of the signals from the next channel begins before DMA transfer of conversion results is completed.

  (Examples are when another DMA transfer is activated and DMA transfer of conversion results in progress is made to wait)

For this reason, if the FIFO is storing more data corresponding to stages exceeding the number of stages at which an interrupt request is generated, a clear operation by the DMA controller (DMAC) is ignored and DMA transfer is performed again.

Figure 26.6-7 shows the DMA retransfer operation.

Figure 26.6-7 DMA retransfer operation



CHAPTER 27 8-bit D/A Converter

This chapter explains the functions and operations of the 8-bit D/A converter.

27.1 Overview

27.2 Configuration

27.3 Pins

27.4 Registers

27.5 Explanation of Operations and Setting Procedure Examples

27.1 Overview

The 8-bit D/A converter acts as a peripheral function for converting digital signals into analog signals. This series microcontroller has three built-in channels for the 8-bit D/A converter.

■ Overview

- Powerdown function

  The powerdown function reduces power requirements when output from the D/A converter is disabled.
- Independent control of channels

  Output from the three channels of the D/A converters can be controlled independently of one another.

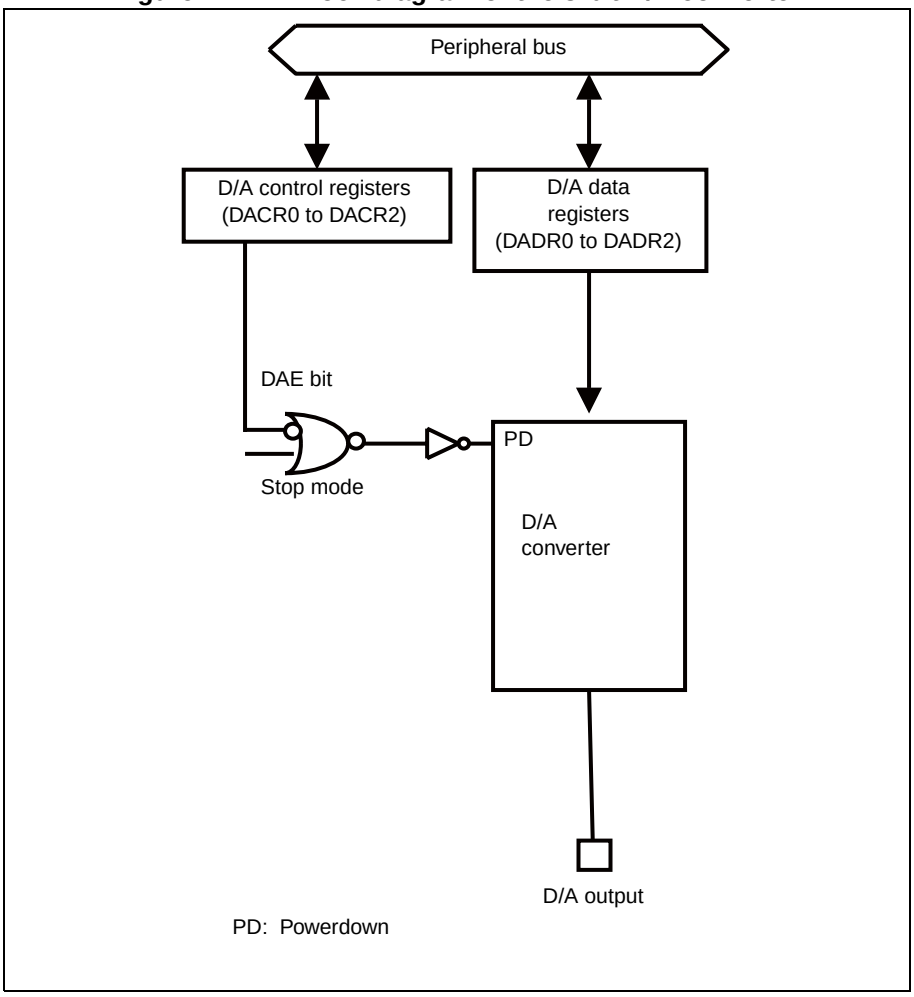
27.2 Configuration

This section explains the configuration of the 8-bit D/A converter.

■ Block diagram of the 8-bit D/A converter

Figure 27.2-1 shows a block diagram of the 8-bit D/A converter.

Figure 27.2-1 Block diagram of the 8-bit D/A converter



- D/A control registers (DACR0 to DACR2)
  These registers control output from the 8-bit D/A converter.
- D/A data registers (DADR0 to DADR2)
  These registers are used to set the output voltages of the D/A converter.
- 8-bit D/A converter
  It converts digital values into analog values.

■ Clocks

Table 27.2-1 lists the clock used for the 8-bit D/A converter.

Table 27.2-1 Clock used for the 8-bit D/A converter

Clock name	Description
Operation clock	Peripheral clock (PCLK)

27.3 Pins

This section explains the pins of the 8-bit D/A converter.

■ Overview

The 8-bit D/A converter has the following pins:

- DA0 to DA2 pins

  Analog output pins of the 8-bit D/A converter.

  These pins are multiplexed pins. For details of using the DA0 to DA2 pins of the 8-bit D/A converter, see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 27.3-1 shows the relationship between channels and pins.

Table 27.3-1 Relationship between channels and pins

Channel	Analog Output Pin
0	DA0
1	DA1
2	DA2

27.4 Registers

This section explains the configurations and functions of the registers used for the 8-bit D/A converter.

■ Registers of the 8-bit D/A converter

Table 27.4-1 lists the registers used for the 8-bit D/A converter.

Table 27.4-1 Registers of the 8-bit D/A converter

Channel	Abbreviated Register Name	Register Name	Reference
0	DADR0	D/A data register 0	27.4.1
	DACR0	D/A control register 0	27.4.2
1	DADR1	D/A data register 1	27.4.1
	DACR1	D/A control register 1	27.4.2
2	DADR2	D/A data register 2	27.4.1
	DACR2	D/A control register 2	27.4.2

27.4.1 D/A Data Registers (DADR0 to DADR2)

These registers are used to set the output voltages from the DA0 to DA2 pins. Based on the values stored in these registers, the values of output voltages from the D/A converter are calculated.

Figure 27.4-1 shows the bit configuration of the D/A data registers (DADR0 to DADR2).

Figure 27.4-1 Bit configuration of the D/A data registers (DADR0 to DADR2)

bit	7	6	5	4	3	2	1	0
	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	X	X	X	X	X	X

R/W: Read/Write

X: Undefined

Table 27.4-2 shows the relationship between the values set in these registers and output voltages.

Table 27.4-2 Setting values and output voltages

DA7 to DA0	Output Voltage
0000 0000	$0/256 \times AV_{CC}$
0000 0001	$1/256 \times AV_{CC}$
0000 0010	$2/256 \times AV_{CC}$
...	...
1111 1101	$253/256 \times AV_{CC}$
1111 1110	$254/256 \times AV_{CC}$
1111 1111	$255/256 \times AV_{CC}$

AV_{CC}: Input voltage from the AV_{CC} pin

<Note>

Even if this product is reset, these registers are not initialized.

27.4.2 D/A Control Registers (DACR0 to DACR2)

These registers control output from the 8-bit D/A converter.

Figure 27.4-2 shows the bit configuration of the D/A control registers (DACR0 to DACR2).

Figure 27.4-2 Bit configuration of the D/A control registers (DACR0 to DACR2)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	DAE
Attribute	-	-	-	-	-	-	-	R/W
Initial value	X	X	X	X	X	X	X	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7 to bit1]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit0]: DAE (D/A output enable bit)

This bit enables/disables output from the 8-bit D/A converter.

Written Value	Explanation
0	Disables output from the D/A converter.
1	Enables output from the D/A converter.

27.5 Explanation of Operations and Setting Procedure Examples

This section explains the operations of the 8-bit D/A converter. Also, examples of procedures for setting the operating state are shown.

27.5.1 Operations of the 8-bit D/A Converter

The 8-bit D/A converter determines the output voltage based on the value that is written to a D/A data register (DADR0 to DADR2), and it outputs an analog voltage through the corresponding DA pin (DA0 to DA2).

When values are written to the DA7 to DA0 bits of a D/A data register (DADR0 to DADR2) and then "1" is written to the DAE bit of a D/A control register (DACR0 to DACR2), the 8-bit D/A converter outputs an analog signal.

The D/A converter outputs 0.0 V when "0" is written to the DAE bit of the D/A control register (DACR0 to DACR2). The D/A converter outputs 0.0 V when "0" is written to the DAE bit, even if the CPU is in stop mode.

<Notes>

- The AV_{CC} pin is shared with the 10-bit A/D converter.
- This D/A converter does not have a built-in buffer amplifier. For details of the electrical characteristics, see "Data Sheet".

CHAPTER 28 Multi-function Serial Interface

This chapter describes the functions and operations of the multi-function serial interface.

28.1 Characteristics of Multi-function Serial Interface
28.2 UART (Asynchronous Serial Interface)
28.3 Overview of UART (Asynchronous Serial Interface)
28.4 Registers of UART (Asynchronous Serial Interface)
28.5 Interrupts of UART
28.6 Operation of UART
28.7 Dedicated Baud Rate Generator
28.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
28.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)
28.10 CSIO (Clock Synchronous Serial Interface)
28.11 Overview of CSIO (Clock Synchronous Serial Interface)
28.12 Registers of CSIO (Clock Synchronous Serial Interface)
28.13 Interrupts of CSIO (Clock Synchronous Serial Interface)
28.14 Operation of CSIO (Clock Synchronous Serial Interface)
28.15 Dedicated Baud Rate Generator
28.16 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)
28.17 I^2C Interface
28.18 Overview of I^2C Interface
28.19 Registers of I^2C Interface
28.20 Interrupts of I^2C Interface
28.21 Dedicated Baud Rate Generator

28.1 Characteristics of Multi-function Serial Interface

This multi-function serial interface has the following characteristics.

■ Interface Mode

The following interface modes are selectable for the multi-function serial interface depending on the operation mode settings.

- UART0 (Asynchronous normal serial interface)
- UART1 (Asynchronous multi-processor serial interface)
- CSIO (Clock synchronous serial interface) (SPI can be supported)
- I^2C (I^2C bus interface)

■ Switching the Interface Mode

To communicate through each serial interface, the serial mode registers (SMR) shown in Table 28.1-1 should be used to set the operation mode before starting the communication.

Table 28.1-1 Switching Interface Mode

MD2	MD1	MD0	Interface mode
0	0	0	UART0 (Asynchronous normal serial interface)
0	0	1	UART1 (Asynchronous multi-processor serial interface)
0	1	0	CSIO (Clock synchronization serial interface) (SPI can be supported)
1	0	0	I^2C (I^2C bus interface)

Note: Settings other than above are prohibited.

<Notes>

- Transmission and reception cannot be guaranteed when the operation mode is switched while one of the serial interfaces is still in use for transmission or reception operation.
- The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

■ Number of Channels

This product has 12 built-in channels for multi-function serial interface. There is no I^2C function for ch.0.

■ Transmission/Reception FIFO

This UART has a 16-byte transmission FIFO and 16-byte reception FIFO. The FIFO steps should be converted to 16 bytes when reading through this text.

There is no FIFO between ch.0 and ch.7.

28.2 UART (Asynchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes those supported in operation modes 0 and 1.

- UART (Asynchronous Serial Interface)
- Overview of UART (Asynchronous Serial Interface)
- Registers of UART (Asynchronous Serial Interface)
  - Serial Control Register (SCR)
  - Serial Mode Register (SMR)
  - Serial Status Register (SSR)
  - Extended Serial Control Register (ESCR)
  - Reception Data Register/Transmission Data Register (RDR/TDR)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of UART
  - Occurrence of Reception Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
  - Occurrence of Transmission Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of UART
- Dedicated Baud Rate Generator
  - Setting Baud Rate
- Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)
- Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

28.3 Overview of UART (Asynchronous Serial Interface)

UART (asynchronous serial interface) is a general-purpose serial data communication interface to perform asynchronous communication (start-stop synchronization) with an external unit. The UART supports a two-way communication function (normal mode) and a master/slave communication function (multi-processor mode: the master and slaves both supported). The UART also has transmission/reception FIFO.

■ Functions of UART (Asynchronous Serial Interface)

		Function
1	Data	• Full-duplex double buffer (when FIFO is not used) • Transmission/reception FIFO (maximum size: 16 bytes each) (when FIFO is used)*1
2	Serial input	Oversampling is performed for three times to determine the reception value by the majority of the sampling values achieved.
3	Transfer system	Asynchronous
4	Baud rate	• Dedicated baud rate generator (15-bit reload counter configuration) • The reload counter can be used to adjust the external clock input.
5	Data length	5 to 9 bits (in normal mode), 7 or 8 bits (in multi-processor mode)
6	Signaling system	NRZ (Non Return to Zero), inverted NRZ
7	Start bit detection	• Synchronized with the falling edge of a start bit (NRZ) • Synchronized with the rising edge of a start bit (inverted NRZ)
8	Reception error detection	• Framing error • Overrun error • Parity error*2
9	Interrupt request	• Reception interrupt (completion of reception, framing error, overrun error, parity error)*2 • Transmission interrupt (transmission data empty, transmission bus idle) • Transmission FIFO interrupt (when transmission FIFO is empty) • DMA transfer support function for transmission and reception
10	Master/slave communication function (multi-processor mode)	Communication between 1 (master) and n (slaves) is enabled. (The master and slave systems are both supported.)

		Function
11	FIFO options	• Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)*1 • Transmission FIFO or reception FIFO selectable • Transmission data can be resent. • The interrupt timing for reception FIFO can be modified by software. • FIFO reset is supported separately.

*1: There is no FIFO between ch.0 and ch.7.

*2: The detection of a parity error is enabled only in normal mode.

28.4 Registers of UART (Asynchronous Serial Interface)

This section lists the registers of UART (asynchronous serial interface).

■ List of Registers of UART (Asynchronous Serial Interface)

Table 28.4-1 List of Registers of UART (Asynchronous Serial Interface) (1 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
0	SCR0	Serial control register 0	28.4.1
	SMR0	Serial mode register 0	28.4.2
	ESCR0	Extended serial control register 0	28.4.4
	BGR0	Baud rate generator register 0	28.4.6
	SSR0	Serial status register 0	28.4.3
	RDR0	Received data register 0	28.4.5
	TDR0	Transmitted data register 0	28.4.5
1	SCR1	Serial control register 1	28.4.1
	SMR1	Serial mode register 1	28.4.2
	ESCR1	Extended serial control register 1	28.4.4
	BGR1	Baud rate generator register 1	28.4.6
	SSR1	Serial status register 1	28.4.3
	RDR1	Received data register 1	28.4.5
	TDR1	Transmitted data register 1	28.4.5
2	SCR2	Serial control register 2	28.4.1
	SMR2	Serial mode register 2	28.4.2
	ESCR2	Extended serial control register 2	28.4.4
	BGR2	Baud rate generator register 2	28.4.6
	SSR2	Serial status register 2	28.4.3
	RDR2	Received data register 2	28.4.5
	TDR2	Transmitted data register 2	28.4.5

Table 28.4-1 List of Registers of UART (Asynchronous Serial Interface) (2 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
3	SCR3	Serial control register 3	28.4.1
	SMR3	Serial mode register 3	28.4.2
	ESCR3	Extended serial control register 3	28.4.4
	BGR3	Baud rate generator register 3	28.4.6
	SSR3	Serial status register 3	28.4.3
	RDR3	Received data register 3	28.4.5
	TDR3	Transmitted data register 3	28.4.5
4	SCR4	Serial control register 4	28.4.1
	SMR4	Serial mode register 4	28.4.2
	ESCR4	Extended serial control register 4	28.4.4
	BGR4	Baud rate generator register 4	28.4.6
	SSR4	Serial status register 4	28.4.3
	RDR4	Received data register 4	28.4.5
	TDR4	Transmitted data register 4	28.4.5
5	SCR5	Serial control register 5	28.4.1
	SMR5	Serial mode register 5	28.4.2
	ESCR5	Extended serial control register 5	28.4.4
	BGR5	Baud rate generator register 5	28.4.6
	SSR5	Serial status register 5	28.4.3
	RDR5	Received data register 5	28.4.5
	TDR5	Transmitted data register 5	28.4.5
6	SCR6	Serial control register 6	28.4.1
	SMR6	Serial mode register 6	28.4.2
	ESCR6	Extended serial control register 6	28.4.4
	BGR6	Baud rate generator register 6	28.4.6
	SSR6	Serial status register 6	28.4.3
	RDR6	Received data register 6	28.4.5
	TDR6	Transmitted data register 6	28.4.5

Table 28.4-1 List of Registers of UART (Asynchronous Serial Interface) (3 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
7	SCR7	Serial control register 7	28.4.1
	SMR7	Serial mode register 7	28.4.2
	ESCR7	Extended serial control register 7	28.4.4
	BGR7	Baud rate generator register 7	28.4.6
	SSR7	Serial status register 7	28.4.3
	RDR7	Received data register 7	28.4.5
	TDR7	Transmitted data register 7	28.4.5
8	SCR8	Serial control register 8	28.4.1
	SMR8	Serial mode register 8	28.4.2
	ESCR8	Extended serial control register 8	28.4.4
	BGR8	Baud rate generator register 8	28.4.6
	SSR8	Serial status register 8	28.4.3
	RDR8	Received data register 8	28.4.5
	TDR8	Transmitted data register 8	28.4.5
	FCR18	FIFO control register 18	28.4.7
	FCR08	FIFO control register 08	28.4.8
	FBYTE18	FIFO1 byte register 8	28.4.9
	FBYTE28	FIFO2 byte register 8	28.4.9
9	SCR9	Serial control register 9	28.4.1
	SMR9	Serial mode register 9	28.4.2
	ESCR9	Extended serial control register 9	28.4.4
	BGR9	Baud rate generator register 9	28.4.6
	SSR9	Serial status register 9	28.4.3
	RDR9	Received data register 9	28.4.5
	TDR9	Transmitted data register 9	28.4.5
	FCR19	FIFO control register 19	28.4.7
	FCR09	FIFO control register 09	28.4.8
	FBYTE19	FIFO1 byte register 9	28.4.9
	FBYTE29	FIFO2 byte register 9	28.4.9

Table 28.4-1 List of Registers of UART (Asynchronous Serial Interface) (4 / 4)

Channel	Abbreviated Register Name	Register Name	Reference
10	SCR10	Serial control register 10	28.4.1
	SMR10	Serial mode register 10	28.4.2
	ESCR10	Extended serial control register 10	28.4.4
	BGR10	Baud rate generator register 10	28.4.6
	SSR10	Serial status register 10	28.4.3
	RDR10	Received data register 10	28.4.5
	TDR10	Transmitted data register 10	28.4.5
	FCR110	FIFO control register 110	28.4.7
	FCR010	FIFO control register 010	28.4.8
	FBYTE110	FIFO1 byte register 10	28.4.9
	FBYTE210	FIFO2 byte register 10	28.4.9
11	SCR11	Serial control register 11	28.4.1
	SMR11	Serial mode register 11	28.4.2
	ESCR11	Extended serial control register 11	28.4.4
	BGR11	Baud rate generator register 11	28.4.6
	SSR11	Serial status register 11	28.4.3
	RDR11	Received data register 11	28.4.5
	TDR11	Transmitted data register 11	28.4.5
	FCR111	FIFO control register 111	28.4.7
	FCR011	FIFO control register 011	28.4.8
	FBYTE111	FIFO1 byte register 11	28.4.9
	FBYTE211	FIFO2 byte register 11	28.4.9

Table 28.4-2 Bit Assignment of UART (Asynchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/SMR	UPCL	-	-	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SBL	BDS	SCKE	SOE
SSR/ ESCR	REC	-	PE	FRE	ORE	RDRF	TDRE	TBI	-	ESBL	INV	PEN	P	L2	L1	L0
RDR/TDR	-							D8 (AD)	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	EXT	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

■ Operation Mode

The UART (asynchronous serial interface) operates in two different modes. The mode selection is determined by MD2, MD1 and MD0 in the serial mode register (SMR).

Table 28.4-3 Operation Modes of UART (Asynchronous Serial Interface)

Operation mode	MD2	MD1	MD0	Type
0	0	0	0	UART0 (asynchronous normal mode)
1	0	0	1	UART1 (asynchronous multi-processor mode)

28.4.1 Serial Control Register (SCR)

The serial control register (SCR) enables or disables transmission/reception, transmission/reception interrupts, and transmission bus idle interrupts. SCR can also reset the UART.

■ Serial Control Register (SCR)

Figure 28.4-1 shows the bit structure of the serial control register (SCR), and Table 28.4-4 describes the function of each bit.

Figure 28.4-1 Bit Structure of Serial Control Register (SCR)

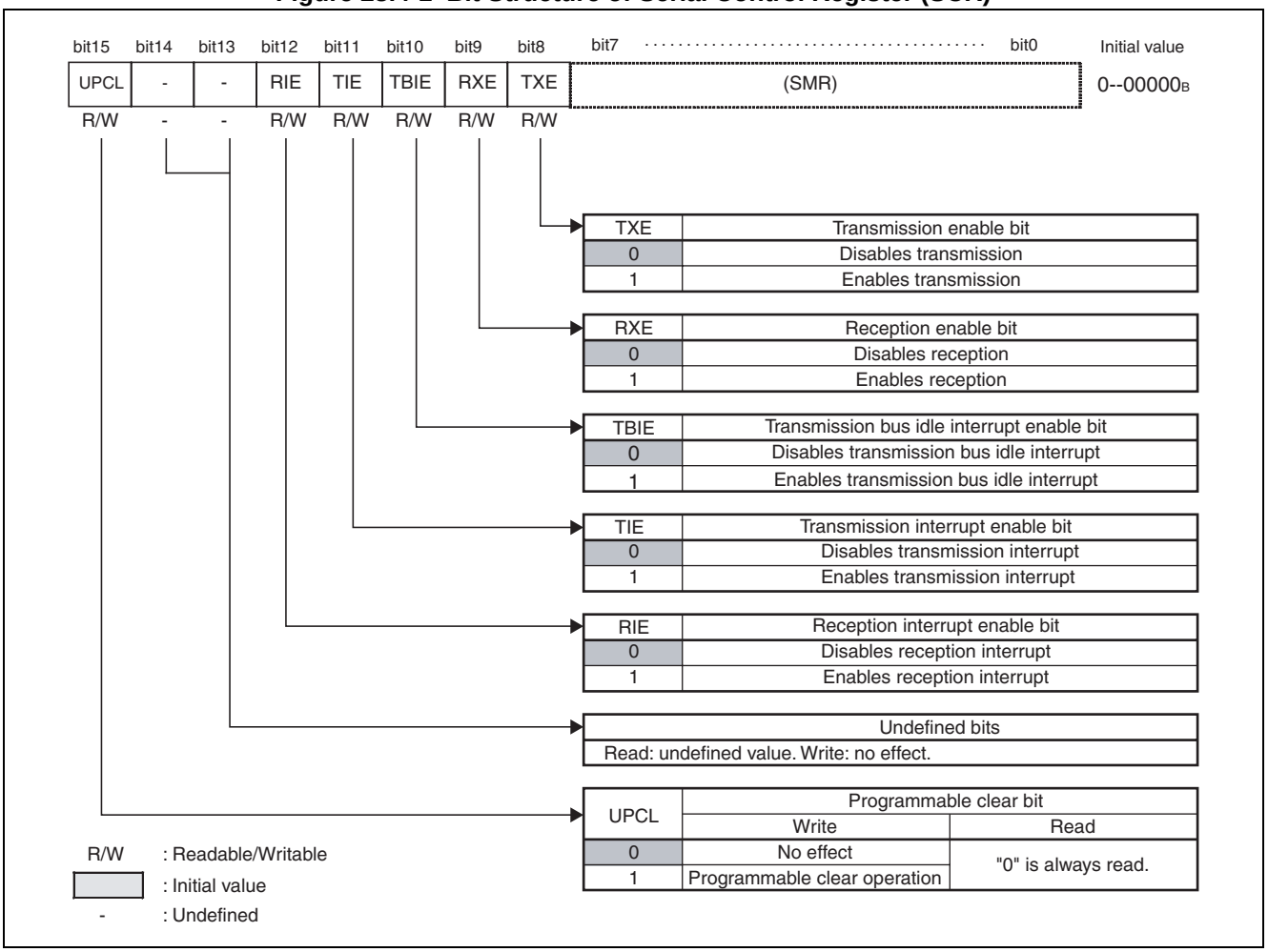

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
UPCL - - RIE TIE TBIE RXE TXE (SMR) 0--00000B
R/W - - R/W R/W R/W R/W R/W
TXE Transmission enable bit
0 Disables transmission
1 Enables transmission
RXE Reception enable bit
0 Disables reception
1 Enables reception
TBIE Transmission bus idle interrupt enable bit
0 Disables transmission bus idle interrupt
1 Enables transmission bus idle interrupt
TIE Transmission interrupt enable bit
0 Disables transmission interrupt
1 Enables transmission interrupt
RIE Reception interrupt enable bit
0 Disables reception interrupt
1 Enables reception interrupt
Undefined bits
Read: undefined value. Write: no effect.
UPCL Programmable clear bit
Write Read
0 No effect "0" is always read.
1 Programmable clear operation
R/W : Readable/Writable
: Initial value
- : Undefined

Table 28.4-4 Functional Description of Each Bit of Serial Control Register (SCR)

Bit name		Function
bit15	UPCL: Programmable clear bit	This bit is used to initialize the internal state of the UART. Setting the bit to "1": • The UART will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. • The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. • All the transmission/reception interrupt sources (PE, FRE, ORE, RDRF, TDRE and TBI) will be initialized (000011_B). Setting the bit to "0": No effect on the operation Reading this bit always returns "0". Note: Execute the programmable clear operation after disabling interrupts. Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.
bit14, bit13	Undefined bits	Read: undefined value Write: no effect
bit12	RIE: Reception interrupt enable bit	• This bit is used to enable/disable the output of reception interrupt requests to the CPU. • A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (PE, ORE or FRE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	• This bit is used to enable/disable the output of transmission interrupt requests to the CPU. • A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	• This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. • A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	This bit is used to enable/disable UART reception operation. • Setting the bit to "0" disables the reception operation. • Setting the bit to "1" enables the reception operation. Note: Even when the reception operation is enabled (RXE = 1), such operation does not start until the falling edge of a start bit (in NRZ format: INV = 0) is input. (When the inverted NRZ format is selected (INV = 1), the reception operation does not start until the rising edge is input.) If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.
bit8	TXE: Transmission enable bit	This bit is used to enable/disable UART transmission operation. • Setting the bit to "0" disables the transmission operation. • Setting the bit to "1" enables the transmission operation. Note: If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.

28.4.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction, data length and stop bit length, and enables or disables the output to the serial data and serial clock pins.

■ Serial Mode Register (SMR)

Figure 28.4-2 shows the bit structure of the serial mode register (SMR), and Table 28.4-5 describes the function of each bit.

Figure 28.4-2 Bit Structure of Serial Mode Register (SMR)

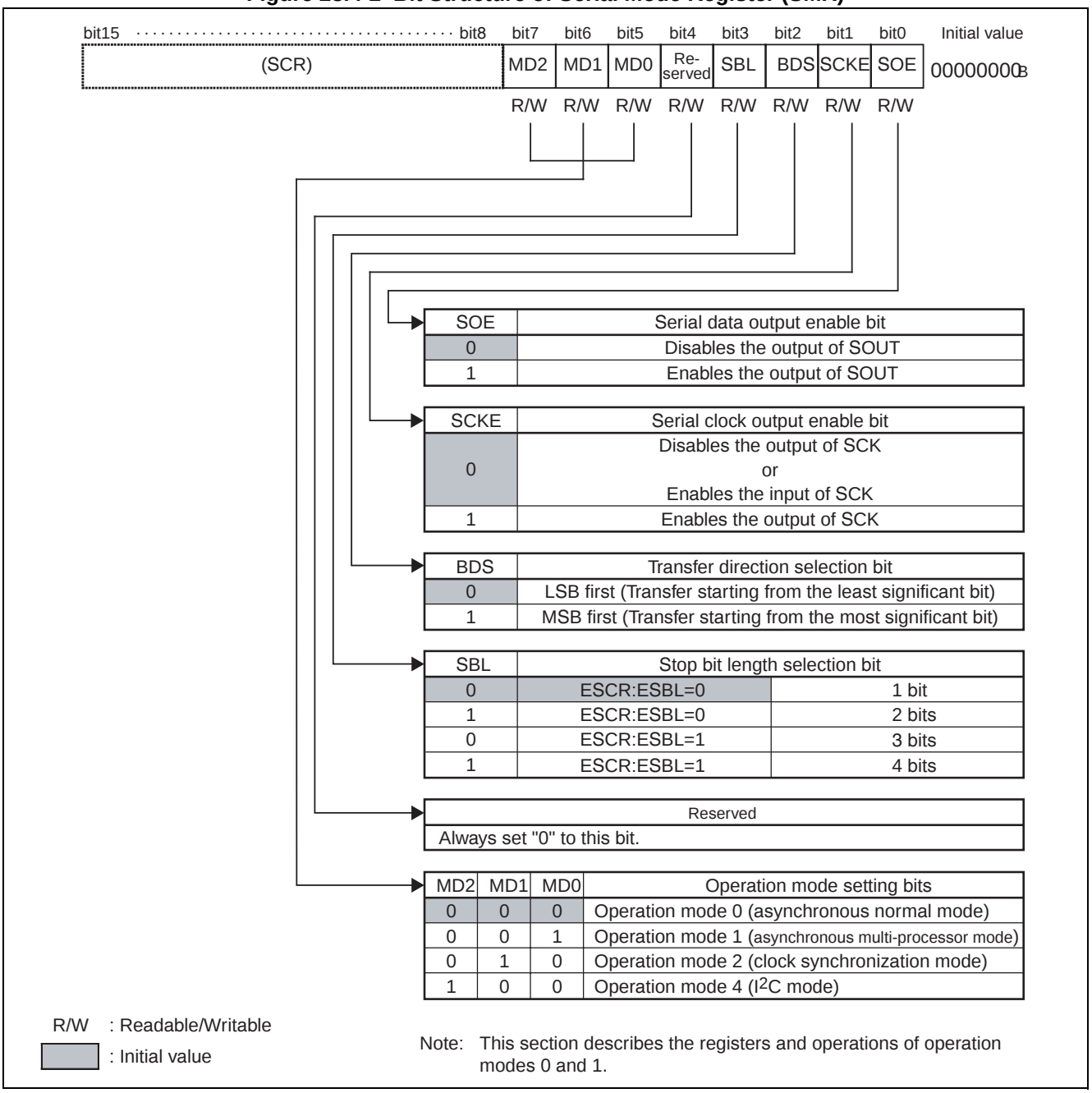

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(SCR) MD2 MD1 MD0 Re-served SBL BDS SCKE SOE 00000000B
R/W R/W R/W R/W R/W R/W R/W R/W
SOE Serial data output enable bit
0 Disables the output of SOUT
1 Enables the output of SOUT
SCKE Serial clock output enable bit
0 Disables the output of SCK or Enables the input of SCK
1 Enables the output of SCK
BDS Transfer direction selection bit
0 LSB first (Transfer starting from the least significant bit)
1 MSB first (Transfer starting from the most significant bit)
SBL Stop bit length selection bit
0 ESCR:ESBL=0 1 bit
1 ESCR:ESBL=0 2 bits
0 ESCR:ESBL=1 3 bits
1 ESCR:ESBL=1 4 bits
Reserved
Always set "0" to this bit.
MD2 MD1 MD0 Operation mode setting bits
0 0 0 Operation mode 0 (asynchronous normal mode)
0 0 1 Operation mode 1 (asynchronous multi-processor mode)
0 1 0 Operation mode 2 (clock synchronization mode)
1 0 0 Operation mode 4 (I2C mode)
R/W : Readable/Writable
: Initial value
Note: This section describes the registers and operations of operation modes 0 and 1.

Table 28.4-5 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2, MD1, MD0: Operation mode setting bits	These bits set the operation mode for the asynchronous serial interface. "000_B": Selects operation mode 0 (asynchronous normal mode) "001_B": Selects operation mode 1 (asynchronous multi-processor mode) "010_B": Selects operation mode 2 (clock synchronization mode) "100_B": Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 0 (asynchronous normal mode) and operation mode 1 (asynchronous multi-processor mode). Note: Settings other than above are prohibited. To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.
bit4	Reserved bit	Always set "0" to this bit.
bit3	SBL: Stop bit length selection bit	This bit is used to select a bit length for a stop bit (frame end mark of transmission data). Setting the bit to SBL=0, ESCR:ESBL=0 sets the stop bit to 1 bit in length. Setting the bit to SBL=1, ESCR:ESBL=0 sets the stop bit to 2 bits in length. Setting the bit to SBL=0, ESCR:ESBL=1 sets the stop bit to 3 bits in length. Setting the bit to SBL=1, ESCR:ESBL=1 sets the stop bit to 4 bits in length. Note: • In reception, only the first bit of each stop bit is always detected. • Set this bit when transmission is disabled (TXE=0).
bit2	BDS: Transfer direction selection bit	This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1). Note: Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit1	SCKE: Serial clock output enable bit	This bit is used to control the I/O port of the serial clock. Setting the bit to "0": The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port. Also select the external clock (BGR:EXT = 1) using the external clock selection bit. Setting the bit to "1" enables the output of SCK.
bit0	SOE: Serial data output enable bit	This bit is used to enable/disable the output of serial data. Setting the bit to "0" disables the output. Setting the bit to "1" enables the output of SOUT.

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

28.4.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

■ Serial Status Register (SSR)

Figure 28.4-3 shows the bit structure of the serial status register (SSR) and Table 28.4-6 describes the function of each bit.

Figure 28.4-3 Bit Structure of Serial Status Register (SSR)

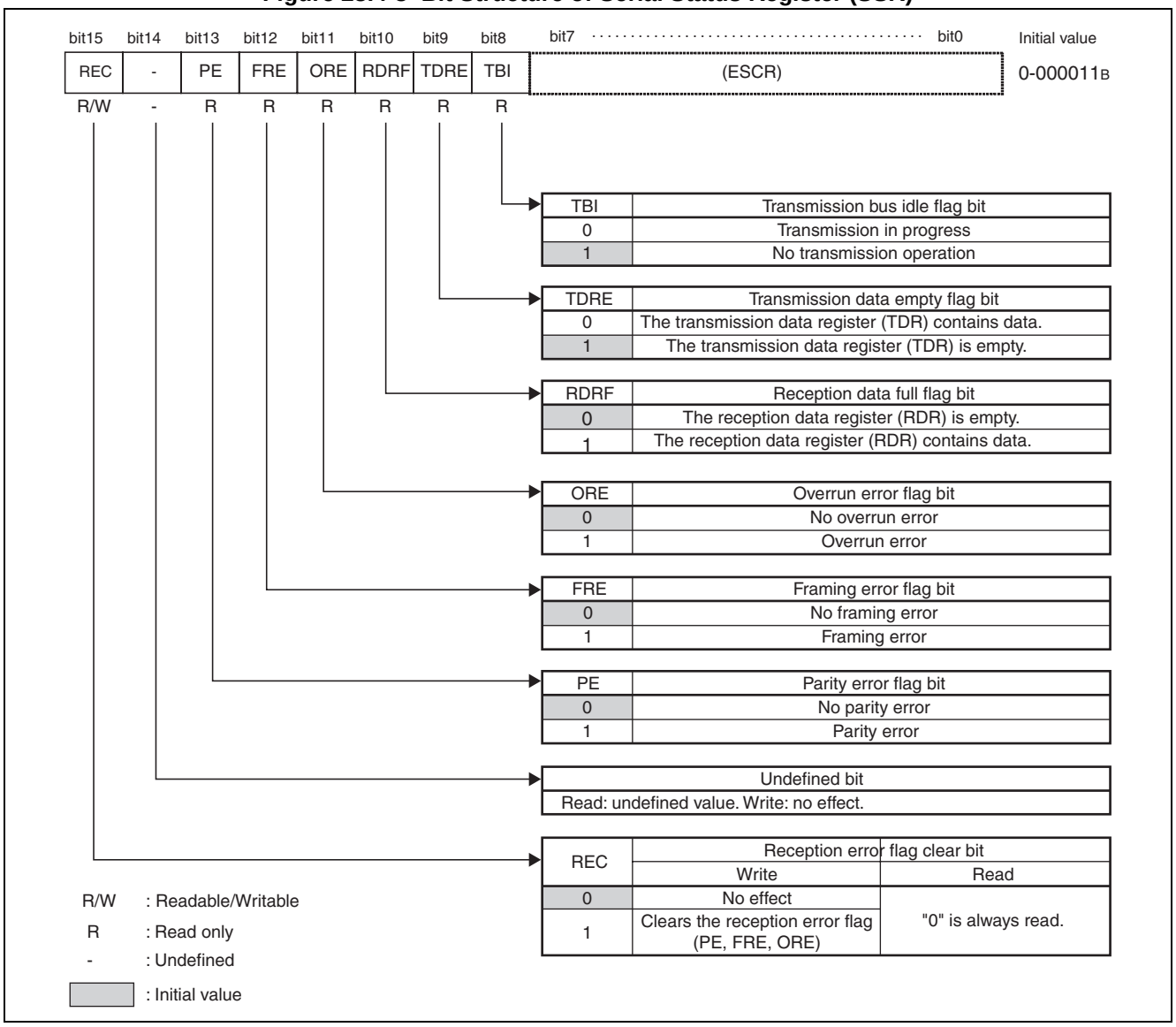

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
REC - PE FRE ORE RDRF TDRE TBI (ESCR) 0-000011B
R/W - R R R R R R
TBI: Transmission bus idle flag bit
0: Transmission in progress
1: No transmission operation
TDRE: Transmission data empty flag bit
0: The transmission data register (TDR) contains data.
1: The transmission data register (TDR) is empty.
RDRF: Reception data full flag bit
0: The reception data register (RDR) is empty.
1: The reception data register (RDR) contains data.
ORE: Overrun error flag bit
0: No overrun error
1: Overrun error
FRE: Framing error flag bit
0: No framing error
1: Framing error
PE: Parity error flag bit
0: No parity error
1: Parity error
Undefined bit
Read: undefined value. Write: no effect.
REC: Reception error flag clear bit
Write
Read
0: No effect
1: Clears the reception error flag (PE, FRE, ORE)
"0" is always read.
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.4-6 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the PE, FRE and ORE flag in the serial status register (SSR). • Writing "1" clears the error flag. • Writing "0" has no effect. Reading this bit always returns "0".
bit14	Undefined bit	Read: undefined value Write: no effect
bit13	PE: Parity error flag bit (only available in operation mode 0)	• The bit is set to "1" when a parity error occurs during reception (ESCR:PEN = 1). The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the PE bit and the SCR:RIE bit are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of reception FIFO, the reception FIFO enable bit will be cleared and no reception data will be stored to the reception FIFO.
bit12	FRE: Framing error flag bit	• This bit is set to "1" when a framing error occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the FRE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit11	ORE: Overrun error flag bit	• This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

Table 28.4-6 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit10	RDRF: Reception data full flag bit	• This flag indicates the status of the reception data register (RDR). • The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. • A reception interrupt request is output when the RDRF and RIE bits are set to "1". • RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. • When the reception FIFO idle detection enable bit (FCR1:FRIIE) is set to "1" during the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. • This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	• This flag indicates the status of the transmission data register (TDR). • When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. • A transmission interrupt request is output when the TDRE and TIE bits are set to "1". • The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "28.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
bit8	TBI: Transmission bus idle flag bit	• This bit indicates that the UART is not performing transmission operation. • The bit is set to "0" when transmission data is written to the transmission data register (TDR). • The bit is set to "1" when the transmission data register is empty (TDRE =1) and no transmission operation is in progress. • The TBI bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

28.4.4 Extended Serial Control Register (ESCR)

The extended serial control register (ESCR) can be used to set the transmission/reception data length, select the stop bit length, enable/disable the parity bit, select the parity bit, and invert the serial data format.

■ Bit Structure of the Extended Serial Control Register (ESCR)

Figure 28.4-4 shows the bit structure of the extended serial control register (ESCR) and Table 28.4-7 describes the function of each bit.

Figure 28.4-4 Bit Structure of Extended Serial Control Register (ESCR)

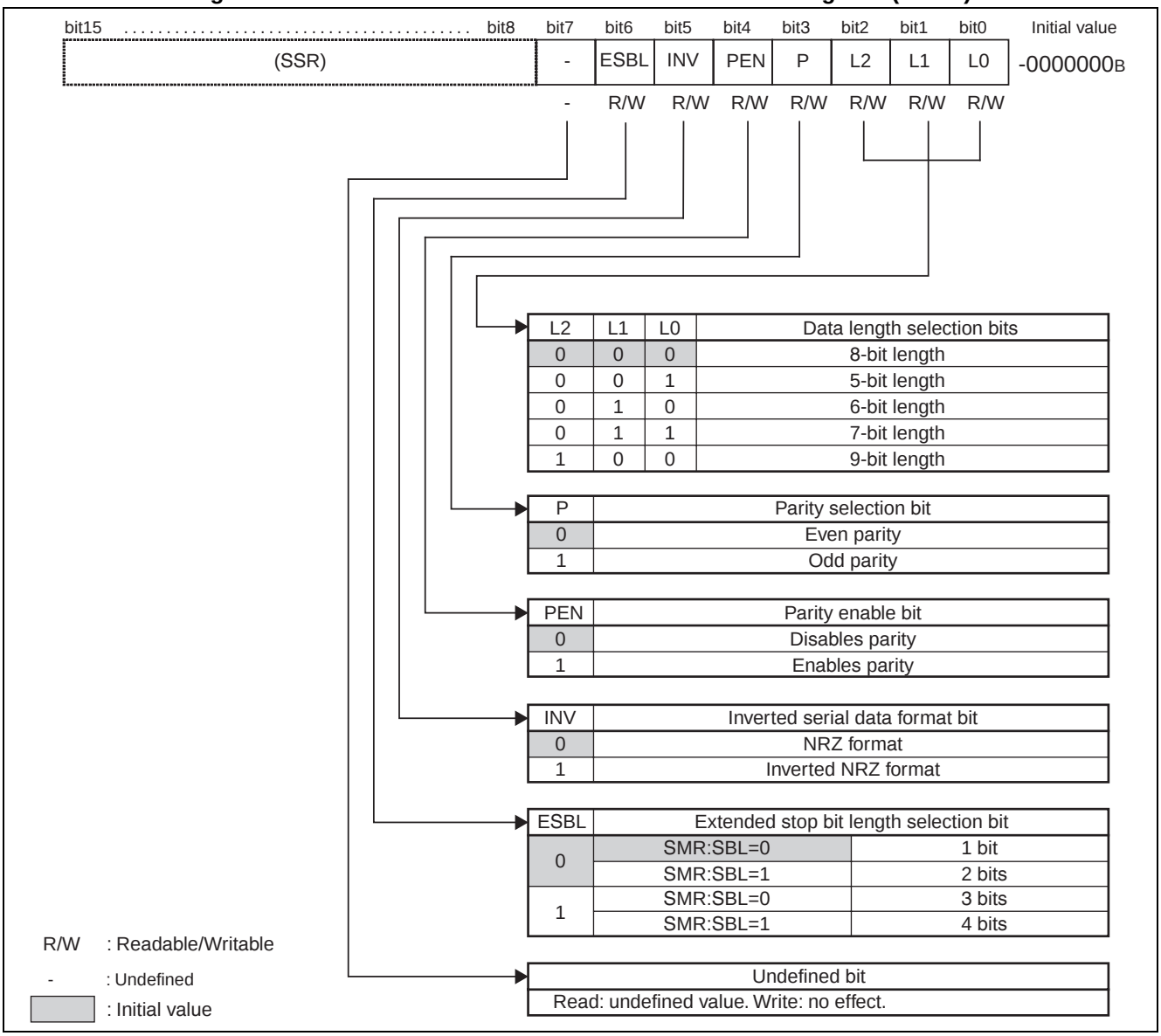

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(SSR) - ESBL INV PEN P L2 L1 L0 -0000000B
- R/W R/W R/W R/W R/W R/W R/W
L2 L1 L0 Data length selection bits
0 0 0 8-bit length
0 0 1 5-bit length
0 1 0 6-bit length
0 1 1 7-bit length
1 0 0 9-bit length
P Parity selection bit
0 Even parity
1 Odd parity
PEN Parity enable bit
0 Disables parity
1 Enables parity
INV Inverted serial data format bit
0 NRZ format
1 Inverted NRZ format
ESBL Extended stop bit length selection bit
0 SMR:SBL=0 1 bit
SMR:SBL=1 2 bits
1 SMR:SBL=0 3 bits
SMR:SBL=1 4 bits
Undefined bit
Read: undefined value. Write: no effect.
R/W : Readable/Writable
- : Undefined
: Initial value

Table 28.4-7 Functional Description of Each Bit of Extended Serial Control Register (ESCR)

Bit name		Function
bit7	Undefined bit	Read: undefined value Write: no effect
bit6	ESBL: Extended stop bit length selection bit	Selects the bit length of stop bit (frame end mark of the transmitted data). Setting SMR: SBL=0, ESBL=0, sets the stop bit to 1 bit. Setting SMR: SBL=1, ESBL=0, sets the stop bit to 2 bits. Setting SMR: SBL=0, ESBL=1, sets the stop bit to 3 bits. Setting SMR: SBL=1, ESBL=1, sets the stop bit to 4 bits. Notes: • Always detects the first bit only of the stop bit during reception. • Set this bit when transmission is prohibited (TXE=0).
bit5	INV: Inverted serial data format bit	This bit is used to select the NRZ format or the inverted NRZ format as the serial data format.
bit4	PEN: Parity enable bit (only available in operation mode 0)	This bit is used to determine whether the parity bit should be added (in transmission) or detected (in reception). • When this bit is set to "0", the parity bit is not added. • When this bit is set to "1", the parity bit is added. Note: This bit is fixed to "0" internally in operation mode 1.
bit3	P: Parity selection bit (only available in operation mode 0)	This bit is used to select odd parity "1" or even parity "0" when parity is enabled (ESCR:PEN = 1). • Setting the bit to "0" selects even parity. • Setting the bit to "1" selects odd parity.
bit2 to bit0	L2, L1, L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. • Selecting "000_B" sets the data length to 8 bits. • Selecting "001_B" sets the data length to 5 bits. • Selecting "010_B" sets the data length to 6 bits. • Selecting "011_B" sets the data length to 7 bits. • Selecting "100_B" sets the data length to 9 bits. Note: Settings other than above are prohibited. For operation mode 1, set the data length to 7 or 8 bits. Any other setting is prohibited.

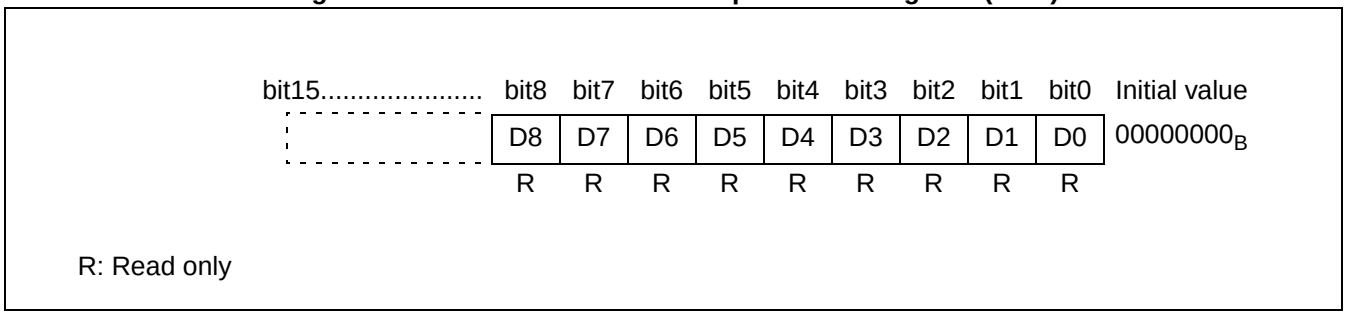
28.4.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.
When FIFO operation is enabled, the RDR/TDR address becomes the read/write address for the FIFO.

■ Reception Data Register (RDR)

Figure 28.4-5 illustrates the bit structure of the serial reception register (RDR).

Figure 28.4-5 Bit Structure of Reception Data Register (RDR)



The reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR).
- "0" is placed in one of the upper bits, depending on the data length, as shown below.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X indicates the reception data bit)

- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the reception data register (RDR) is read.
- If a reception error occurs (one of SSR:PE, ORE, or FRE is "1"), the data in the reception data register (RDR) becomes invalid.
- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the received AD bit is stored in bit D8.
- 16-bit access is used to read RDR for a 9-bit transfer in operation mode 1.

<Notes>

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
- If a reception error occurs (one of SSR:PE, ORE, or FRE is "1") when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

■ Transmission Data Register (TDR)

Figure 28.4-6 illustrates the bit structure of the transmission data register.

Figure 28.4-6 Bit Structure of Transmission Data Register (TDR)

bit15.....	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	11111111_B
	W	W	W	W	W	W	W	W	W	

W: Write only

The transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOUT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

(X indicates the reception data bit)

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- Transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".

- Transmission data cannot be written when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.
- In operation mode 1 (multi-processor mode), 7-bit or 8-bit operation is performed and the AD bit is sent by writing to bit D8.
- 16-bit access is used to write to TDR for a 9-bit transfer in operation mode 1.

<Notes>

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. These registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
- For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "28.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".

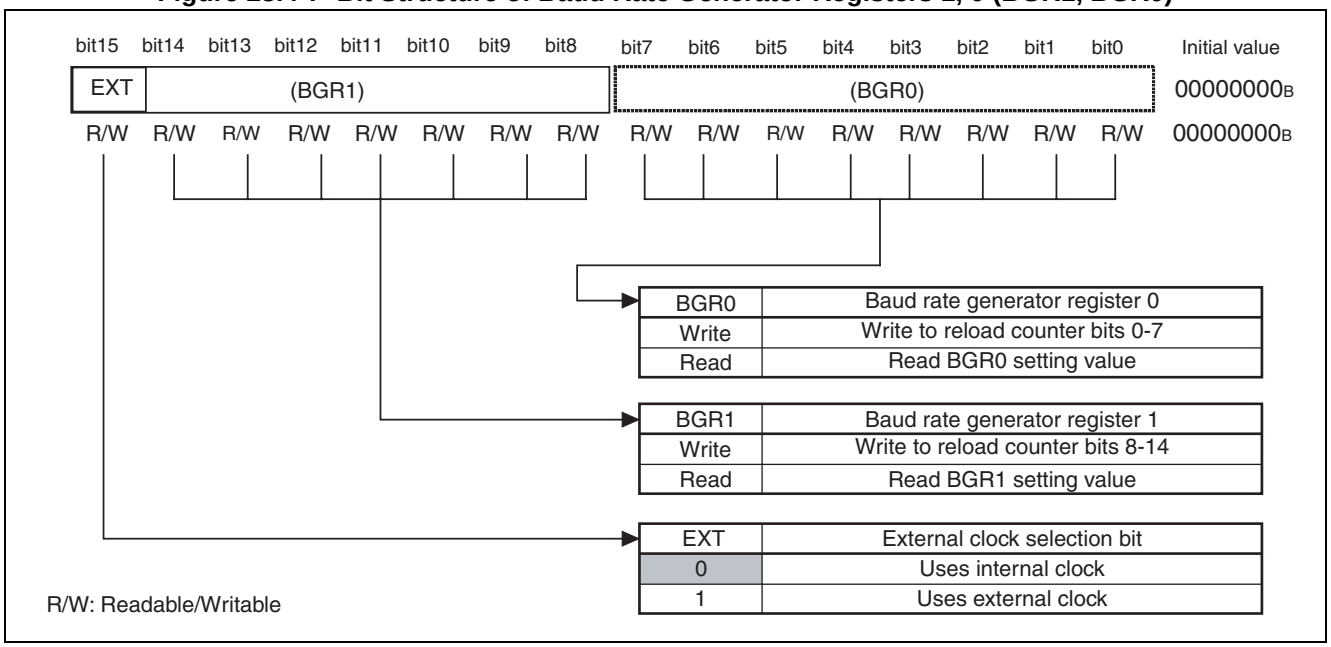
28.4.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock. They also allow an external clock to be selected as the clock source for the reload counter.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 28.4-7 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 28.4-7 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
EXT
(BGR1)
(BGR0)
00000000B
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W 00000000B
BGR0 Baud rate generator register 0
Write Write to reload counter bits 0-7
Read Read BGR0 setting value
BGR1 Baud rate generator register 1
Write Write to reload counter bits 8-14
Read Read BGR1 setting value
EXT External clock selection bit
0 Uses internal clock
1 Uses external clock
R/W: Readable/Writable

- The baud rate generator registers are used to set a division ratio for the serial clock.
- BGR1 and BGR0 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers (BGR1/BGR0).
- The EXT bit (bit15) is used to determine whether the internal clock or external clock should be used as the clock source for the reload counter. Setting EXT to "0" selects the internal clock, while setting EXT to "1" selects the external clock.

<Notes>

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a programmable clear (UPCL) operation after changing the BGR1/BGR0 setting value.
- When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (PCLK) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
- Select 4 or a larger value for BGR1/BGR0. However, data may not be able to be received properly, due to a baud rate error or reload settings.
- To change the setting to the external clock (EXT = 1) during the operation of the baud rate generator, write "0" to baud rate generator registers 1, 0 (BGR1, BGR0), execute a programmable clear (UPCL) operation, and then set to the external clock (EXT = 1).

28.4.7 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 28.4-8 shows the bit structure of the FIFO control register 1 (FCR1) and Table 28.4-8 describes the function of each bit.

Figure 28.4-8 Bit Structure of FIFO Control Register 1 (FCR1)

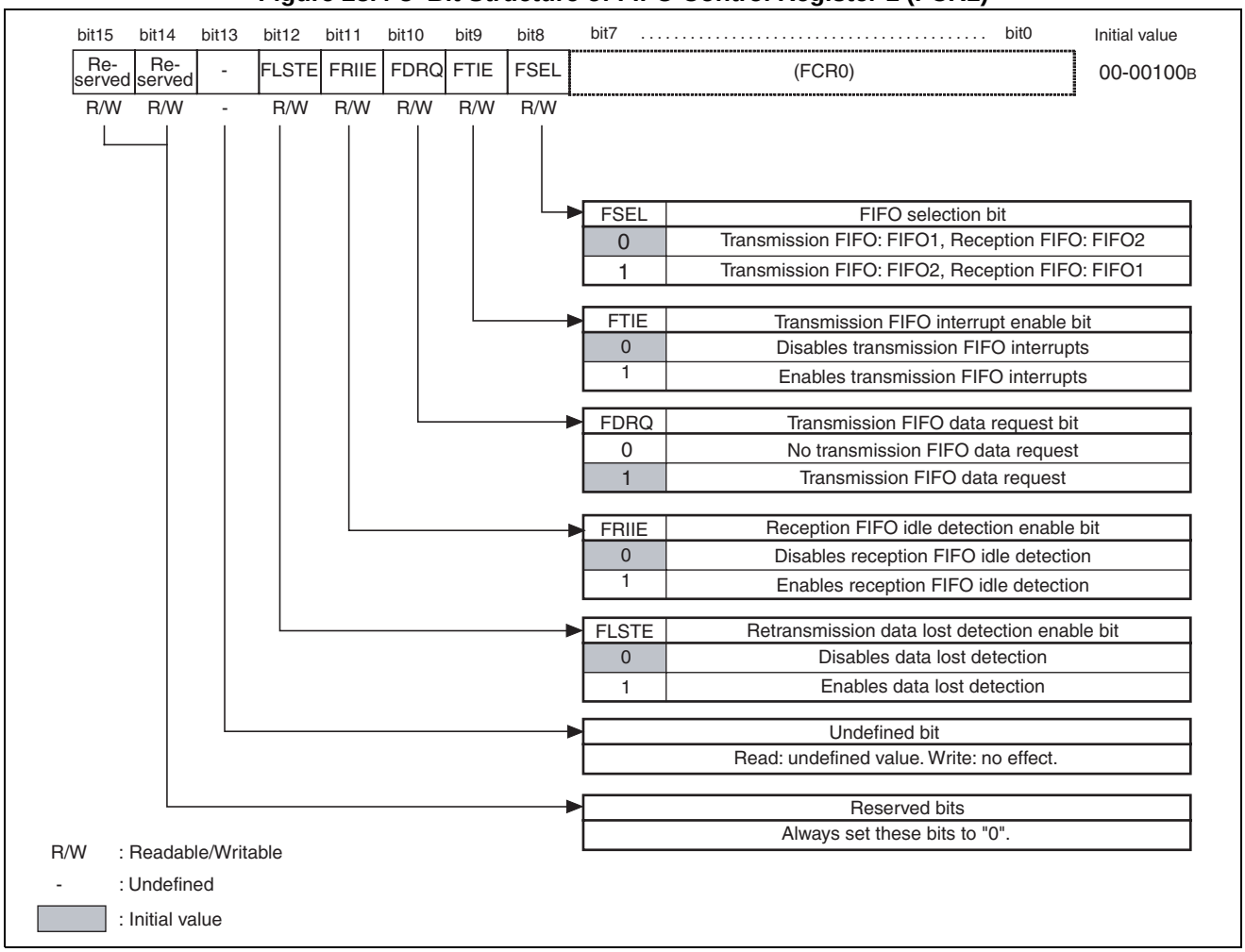

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
Re-served Re-served - FLSTE FRIIE FDRQ FTIE FSEL (FCR0) 00-00100B
R/W R/W - R/W R/W R/W R/W R/W
FSEL FIFO selection bit
0 Transmission FIFO: FIFO1, Reception FIFO: FIFO2
1 Transmission FIFO: FIFO2, Reception FIFO: FIFO1
FTIE Transmission FIFO interrupt enable bit
0 Disables transmission FIFO interrupts
1 Enables transmission FIFO interrupts
FDRQ Transmission FIFO data request bit
0 No transmission FIFO data request
1 Transmission FIFO data request
FRIIE Reception FIFO idle detection enable bit
0 Disables reception FIFO idle detection
1 Enables reception FIFO idle detection
FLSTE Retransmission data lost detection enable bit
0 Disables data lost detection
1 Enables data lost detection
Undefined bit
Read: undefined value. Write: no effect.
Reserved bits
Always set these bits to "0".
R/W : Readable/Writable
- : Undefined
: Initial value

Table 28.4-8 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "0".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) FDRQ reset condition • Writing "0" to this bit • When the transmission FIFO is full. Note: It is valid to write "0" when transmission FIFO has been enabled. It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. Note: This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation beforehand (FCR:FE2, FE1 = 0).

28.4.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 28.4-9 shows the bit structure of the FIFO control register 0 (FCR0) and Table 28.4-9 describes the function of each bit.

Figure 28.4-9 Bit Structure of FIFO Control Register 0 (FCR0)

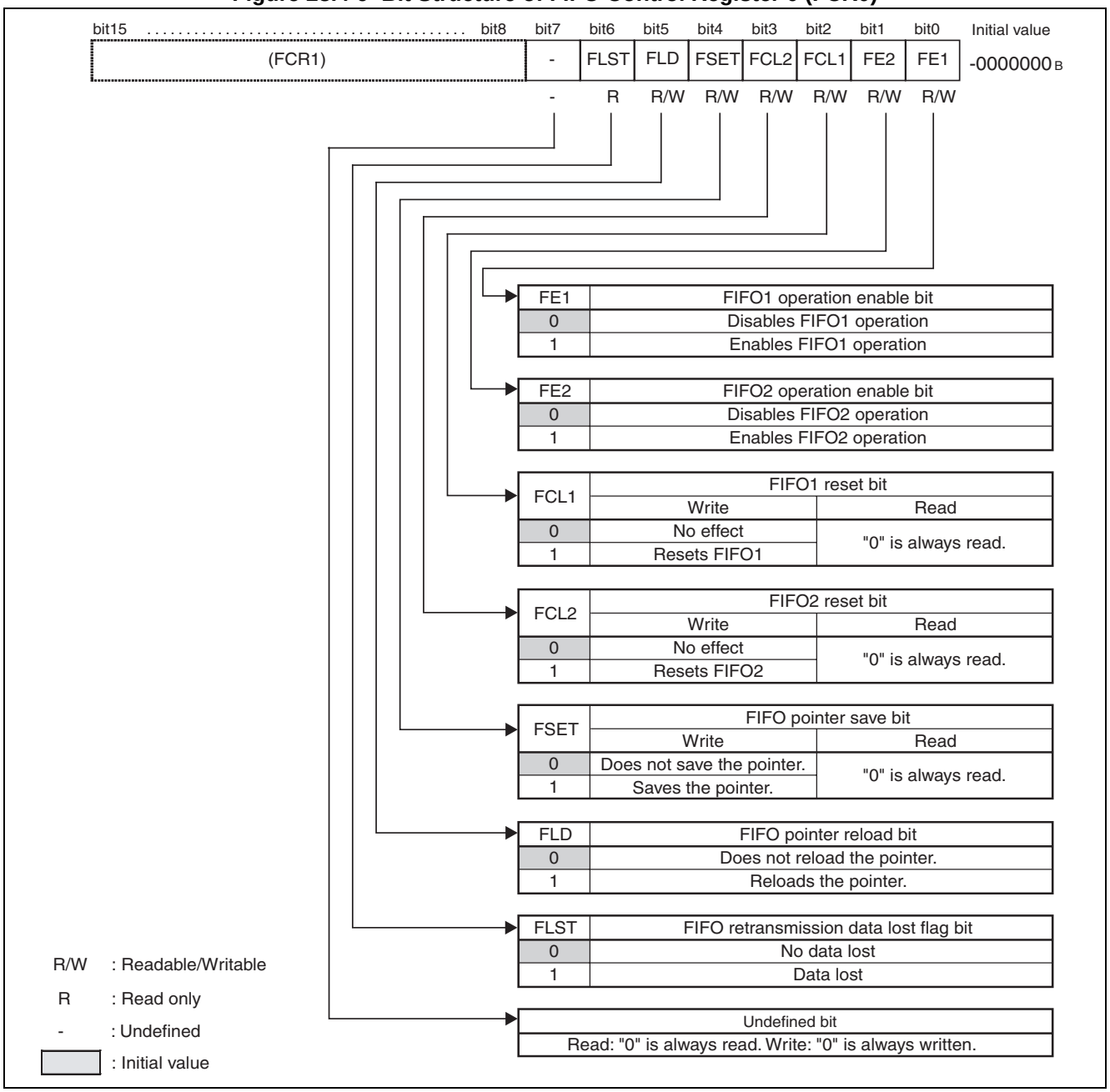

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FCR1) - FLST FLD FSET FCL2 FCL1 FE2 FE1 -0000000B
- R R/W R/W R/W R/W R/W R/W
FE1 FIFO1 operation enable bit
0 Disables FIFO1 operation
1 Enables FIFO1 operation
FE2 FIFO2 operation enable bit
0 Disables FIFO2 operation
1 Enables FIFO2 operation
FCL1 FIFO1 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO1
FCL2 FIFO2 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO2
FSET FIFO pointer save bit
Write Read
0 Does not save the pointer. "0" is always read.
1 Saves the pointer.
FLD FIFO pointer reload bit
0 Does not reload the pointer.
1 Reloads the pointer.
FLST FIFO retransmission data lost flag bit
0 No data lost
1 Data lost
Undefined bit
Read: "0" is always read. Write: "0" is always written.
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.4-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition Writing (overwriting) to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions • FIFO reset (writing "1" to FCL) • Writing "1" to the FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. Note: Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to communication will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. Note: Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR1:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO2. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".

Table 28.4-9 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. • To use FIFO2, set this bit to "1". • When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. • To use FIFO1, set this bit to "1". • When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO1 is disabled, its status is retained.

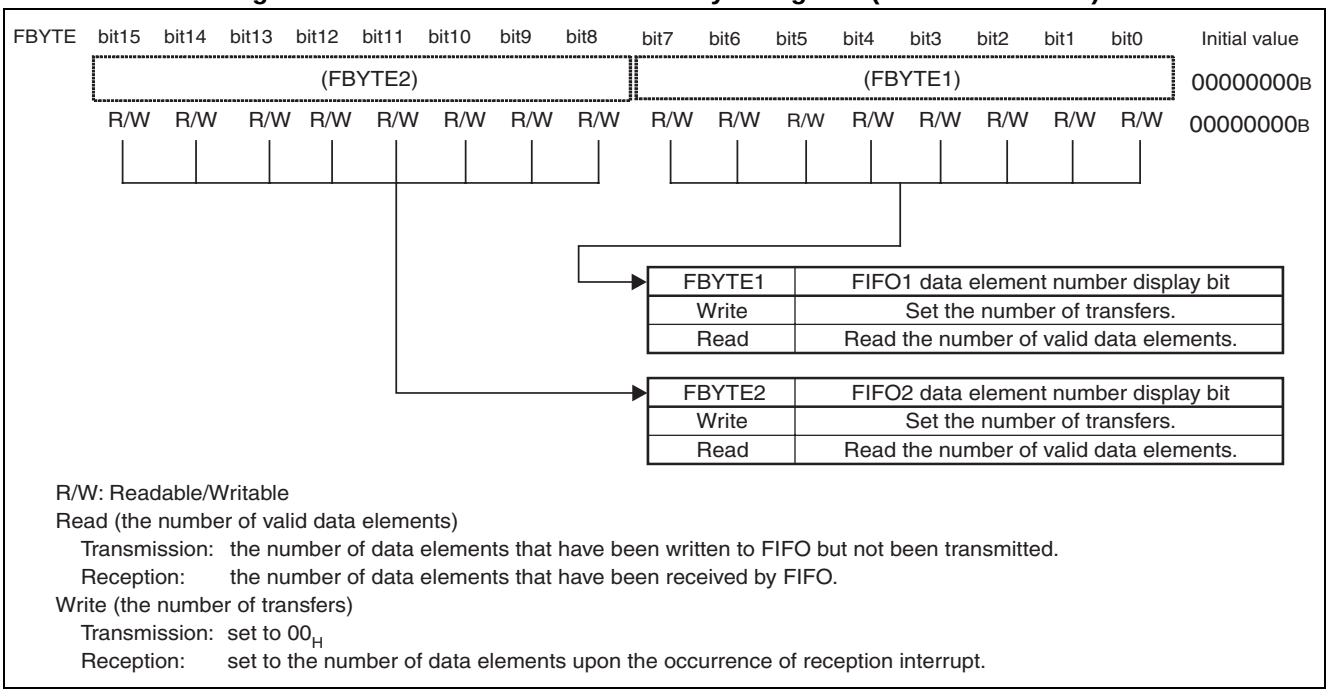
28.4.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. It can also determine whether a reception interrupt should occur when the reception FIFO has received a specified number of data elements.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 28.4-10 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 28.4-10 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)


FBYTE bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FBYTE2)
(FBYTE1)
00000000B
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
00000000B
FBYTE1 FIFO1 data element number display bit
Write Set the number of transfers.
Read Read the number of valid data elements.
FBYTE2 FIFO2 data element number display bit
Write Set the number of transfers.
Read Read the number of valid data elements.

R/W: Readable/Writable
Read (the number of valid data elements)
Transmission: the number of data elements that have been written to FIFO but not been transmitted.
Reception: the number of data elements that have been received by FIFO.
Write (the number of transfers)
Transmission: set to 00_H
Reception: set to the number of data elements upon the occurrence of reception interrupt.

The FBYTE1/FBYTE2 register indicates the number of valid data elements written to or received by FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 28.4-10 Displaying the Number of Data Elements

FSEL	FIFO selection	Number of data elements displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (SSR:RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.

<Notes>

- Set 00_H to the FBYTE1/FBYTE2 register of the transmission FIFO.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- Change this register after prohibiting receiving.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

28.5 Interrupts of UART

The UART has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

■ Interrupts of UART

Table 28.5-1 shows the interrupt control bits and interrupt sources of the UART.

Table 28.5-1 Interrupt Control Bits and Interrupt Sources of UART (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Reception	RDRF	SSR	❍	❍	Reception of 1 byte	SCR:RIE	Reading reception data (RDR)
					Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR) until reception FIFO becomes empty
					Detection of the idle state of reception for 8 clocks with the baud rate or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	❍	❍	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)
	FRE	SSR	❍	❍	Framing error		
	PE	SSR	❍	×	Parity error		

Table 28.5-1 Interrupt Control Bits and Interrupt Sources of UART (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Operation mode		Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
			0	1			
Trans-mission	TDRE	SSR	❍	❍	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	TBI	SSR	❍	❍	No transmission operation	SCR:TBIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	FDRQ	FCR1	❍	❍	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

28.5.1 Occurrence of Reception Interrupts and Flag Set Timing

Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:PE, ORE, FRE).

■ Occurrence of Reception Interrupts and Flag Set Timing

Reception data is stored to the reception data register (RDR) when the first stop bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:PE, ORE, FRE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

<Note>

If a reception error occurs, the data in the reception data register (RDR) will become invalid.

Figure 28.5-1 Timing for Setting RDRF (Reception Data Full) Flag Bit

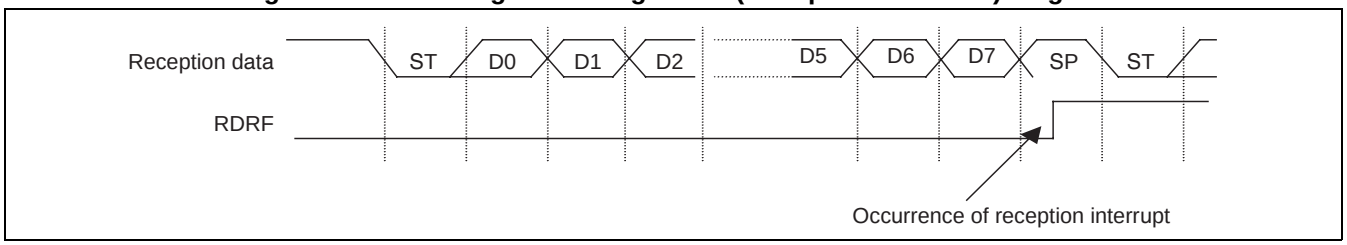
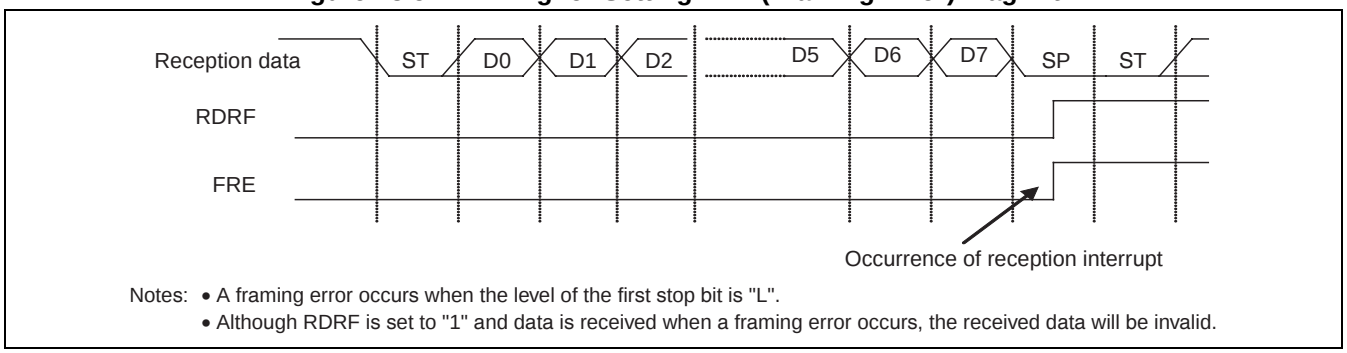
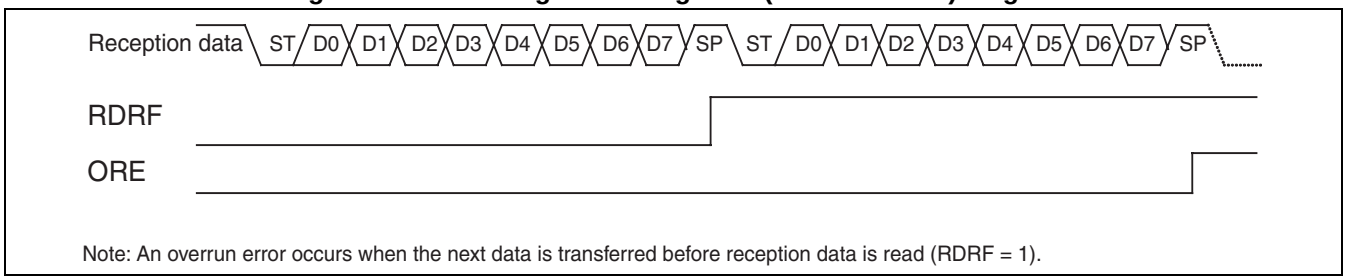


Figure 28.5-2 Timing for Setting FRE (Framing Error) Flag Bit



Notes: • A framing error occurs when the level of the first stop bit is "L".
• Although RDRF is set to "1" and data is received when a framing error occurs, the received data will be invalid.

Figure 28.5-3 Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is transferred before reception data is read (RDRF = 1).

28.5.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1 register (FBYTE1/FBYTE2) is received.

■ Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 28.5-4 Timing for Generating Reception Interrupt when Reception FIFO is Used

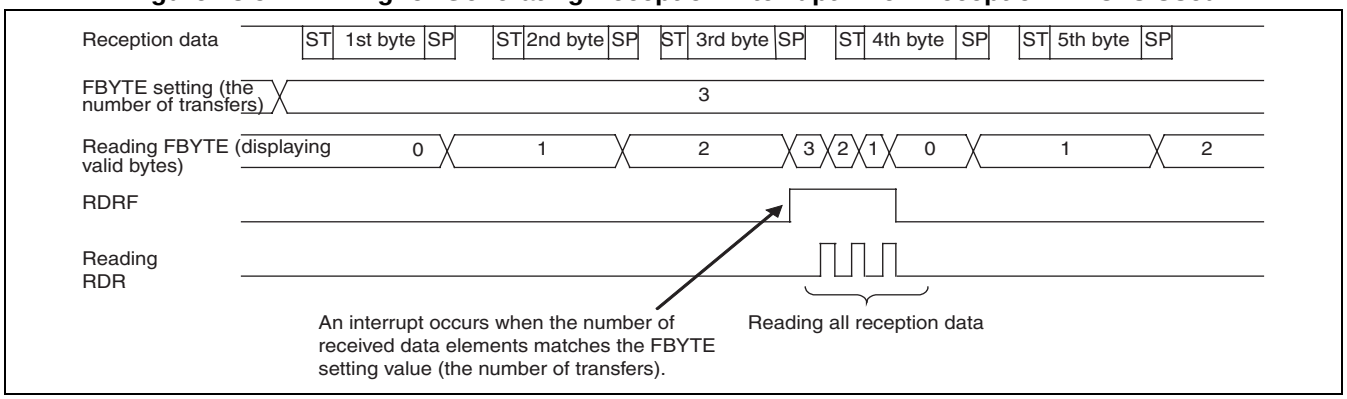
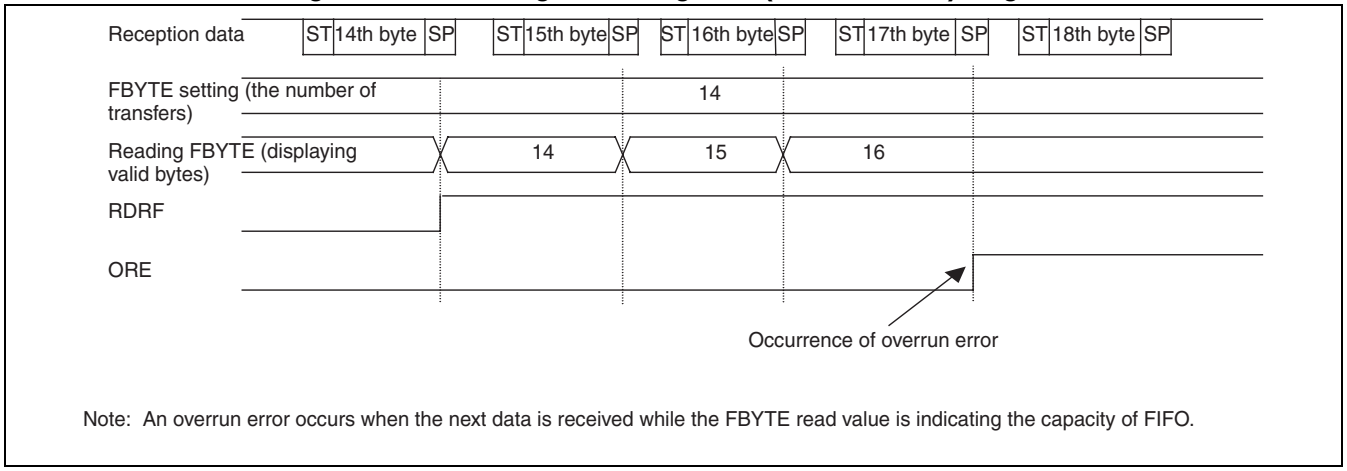


Figure 28.5-5 Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is received while the FBYTE read value is indicating the capacity of FIFO.

28.5.3 Occurrence of Transmission Interrupts and Flag Set Timing

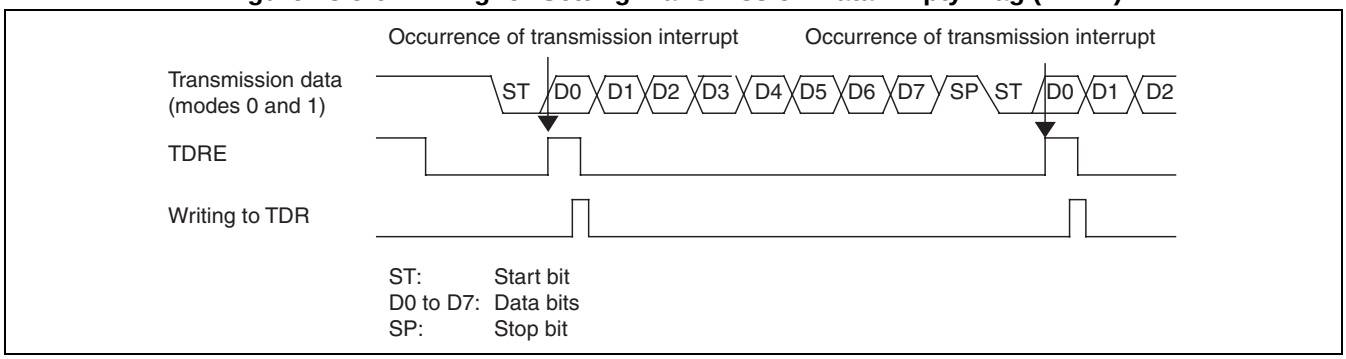
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ Occurrence of Transmission Interrupts and Flag Set Timing

● Timing for setting the transmission data empty flag (TDRE)

It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR).

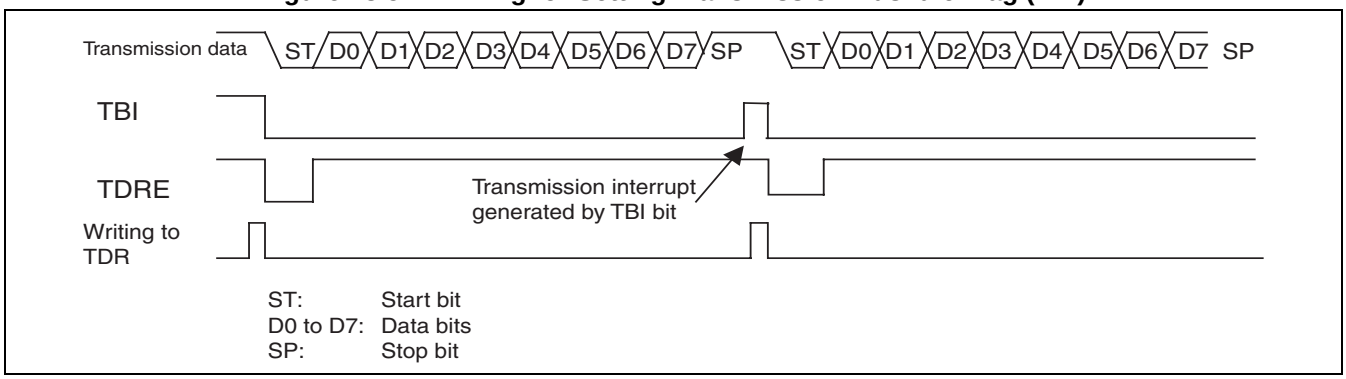
Figure 28.5-6 Timing for Setting Transmission Data Empty Flag (TDRE)



● Timing for setting the transmission bus idle flag (TBI)

The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR).

Figure 28.5-7 Timing for Setting Transmission Bus Idle Flag (TBI)



28.5.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

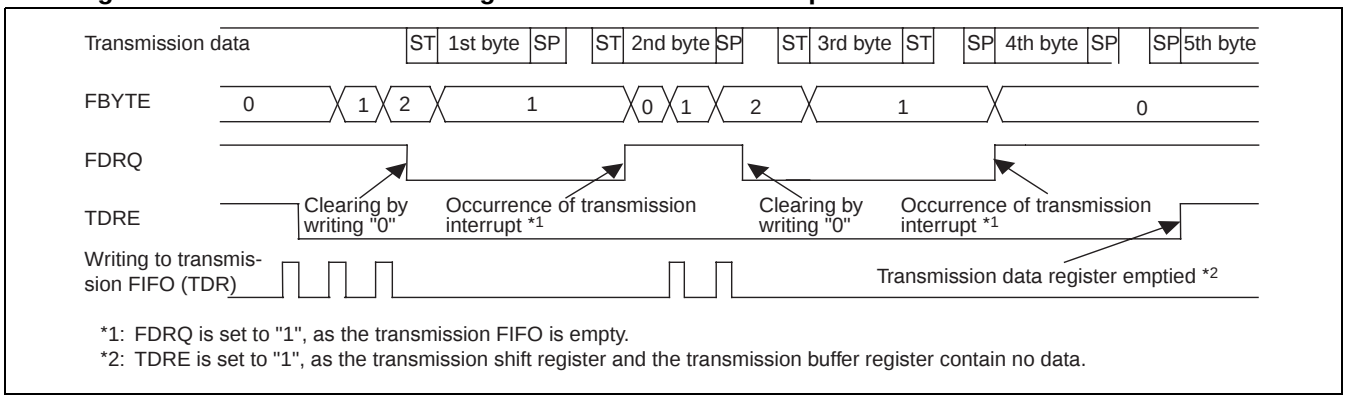
■ Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data.

  At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data.

  FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 28.5-8 Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



*1: FDRQ is set to "1", as the transmission FIFO is empty.
*2: TDRE is set to "1", as the transmission shift register and the transmission buffer register contain no data.

28.6 Operation of UART

The UART operates in two-way serial asynchronous communications for mode 0 and in master/slave multi-processor communications for mode 1.

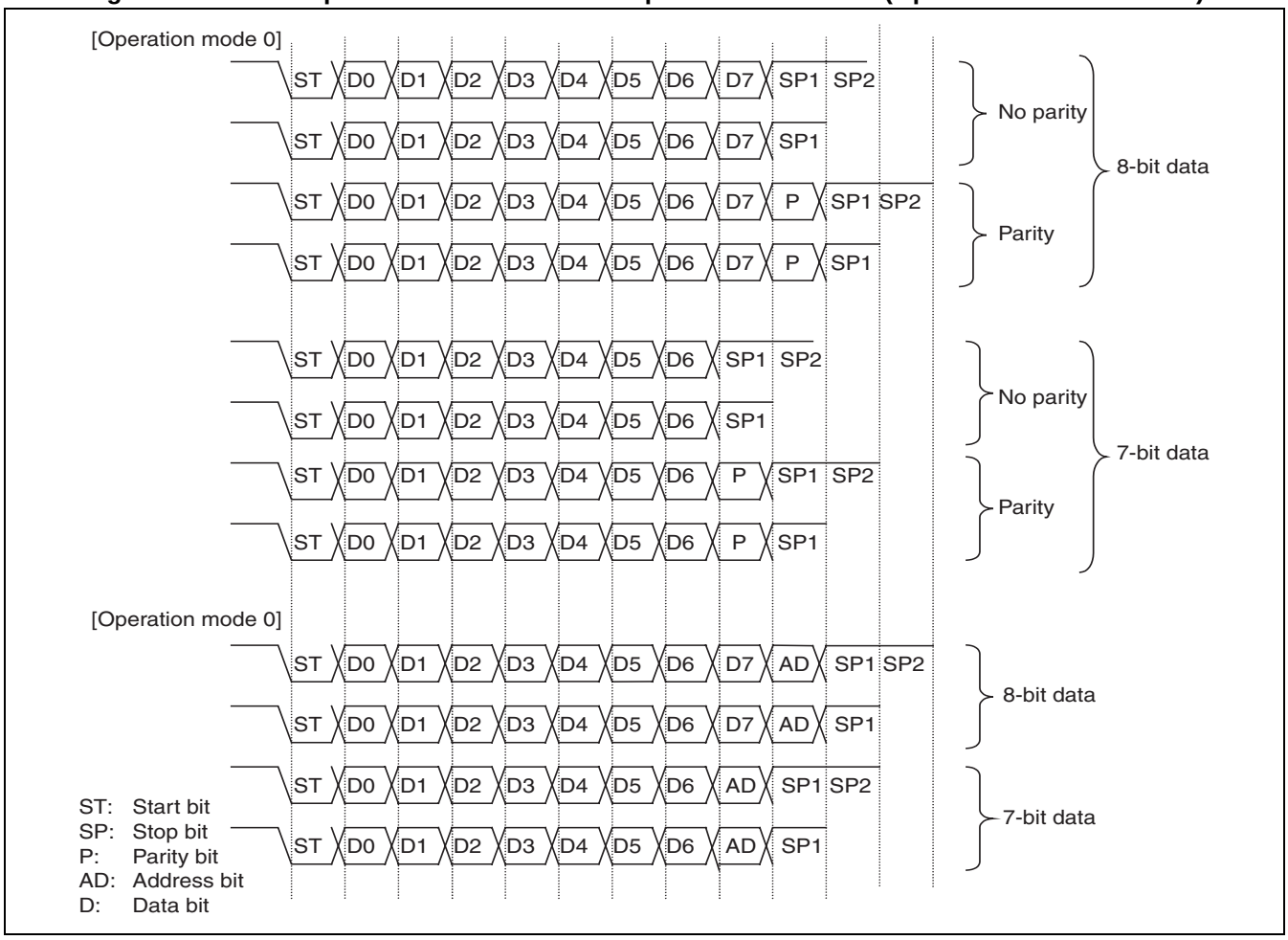
■ Operation of UART

● Transmission/reception data format

- Transmission and reception data is transmitted or received for a specified data bit length, always starting from the start bit and finishing with the stop bit (at least 1 bit).
- The data transfer direction (LSB or MSB first) is determined by the BDS bit in the serial mode register (SMR). When the addition of parity is selected, the parity bit is always placed between the last data bit and the first stop bit.
- The addition or omission of parity can be selected in operation mode 0 (normal mode).
- In operation mode 1 (multi-processor mode), the AD bit is added rather than parity.

Figure 28.6-1 shows transmission/reception data formats for operation modes 0 and 1.

Figure 28.6-1 Examples of Transmission/Reception Data Formats (Operation Modes 0 and 1)


[Operation mode 0]
ST D0 D1 D2 D3 D4 D5 D6 D7 SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 D7 SP1
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP1
No parity
Parity
8-bit data
ST D0 D1 D2 D3 D4 D5 D6 SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 SP1
ST D0 D1 D2 D3 D4 D5 D6 P SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 P SP1
No parity
Parity
7-bit data
[Operation mode 0]
ST D0 D1 D2 D3 D4 D5 D6 D7 AD SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 D7 AD SP1
8-bit data
ST D0 D1 D2 D3 D4 D5 D6 AD SP1 SP2
ST D0 D1 D2 D3 D4 D5 D6 AD SP1
7-bit data
ST: Start bit
SP: Stop bit
P: Parity bit
AD: Address bit
D: Data bit

<Notes>

- Figure 28.6-1 shows cases where the data length is set to 7 or 8 bits. (The data length can be set to 5-9 bits for operation mode 0.)
- When the BDS bit in the serial mode register (SMR) is set to "1" (MSB first), the bits are processed in the following order: D7, D6, D5...D1, D0 (P).
- When the data length is set to X bits, the lower X bits of the transmission/reception data register (RDR/TDR) become valid.

● Transmission operation

- Transmission data can be written to the transmission data register (TDR) when the transmission data empty flag bit (TDRE) in the serial status register (SSR) is set to "1". (If the transmission FIFO is enabled, transmission data can be written even when TDRE is set to "0".)
- Writing transmission data to the transmission data register (TDR) sets the transmission data empty flag bit (TDRE) to "0".
- When the transmission operation enable bit in the serial control register (SCR:TXE) is set to "1", transmission data is loaded to the transmission shift register and the transmission starts from the start bit.
- Once transmission starts, the transmission data empty flag bit (TDRE) is set back to "1". At this point, a transmission interrupt will occur if transmission interrupts have been enabled (SCR:TIE = 1). The next transmission data can be written to the transmission data register through interrupt processing.

<Notes>

- The initial value of the transmission data empty flag bit (SSR:TDRE) is "1". Therefore, a transmission interrupt occurs immediately after transmission interrupts are enabled (SCR:TIE=1).
- The initial value of the FIFO transmission data request bit (FCR1:FDRQ) is "1". Therefore, a transmission interrupt occurs immediately after FIFO transmission interrupts are enabled (FCR1:FTIE = 1).

● Reception operation

- Reception operation starts when such operation is enabled (SCR:RXE = 1).
- When a start bit is detected, one frame of data is received according to the data format set in the extended serial control register (ESCR:PEN, P, L2, L1, L0) and the serial mode register (SMR:BDS).
- Once one frame of data has been received, the reception data full flag bit (SSR:RDRF) is set to "1". At this point, a reception interrupt will occur if reception interrupts have been enabled (SCR:RIE = 1).
- Read reception data after one frame of data has been received, and then check the error flag status of the serial status register (SSR). If a reception error is occurring, the error must be treated.
- Reading reception data clears the reception data full flag bit (SSR:RDRF) to "0".
- When the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be set to "1", if the received data is equivalent of the number of frames specified in the reception FBYTE1/ FBYTE2.
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- If the error flag in the serial status register (SSR) is set to "1" when the reception FIFO has been enabled, the data in which the error has occurred will not be stored to the reception FIFO. At the same time, the reception data full flag bit (SSR:RDRF) will not be set to "1". (In case of an overrun error, however, the RDRF flag will be set to "1".) The reception FBYTE1/FBYTE2 indicates the number of data elements that was successfully received before the error occurs. The reception FIFO will not be enabled unless the error flag in the serial status register (SSR) is cleared to "0".
- If the reception FIFO has been enabled, the reception data full flag bit (SSR:RDRF) will be cleared to "0" when the reception FIFO no longer has data.

<Note>

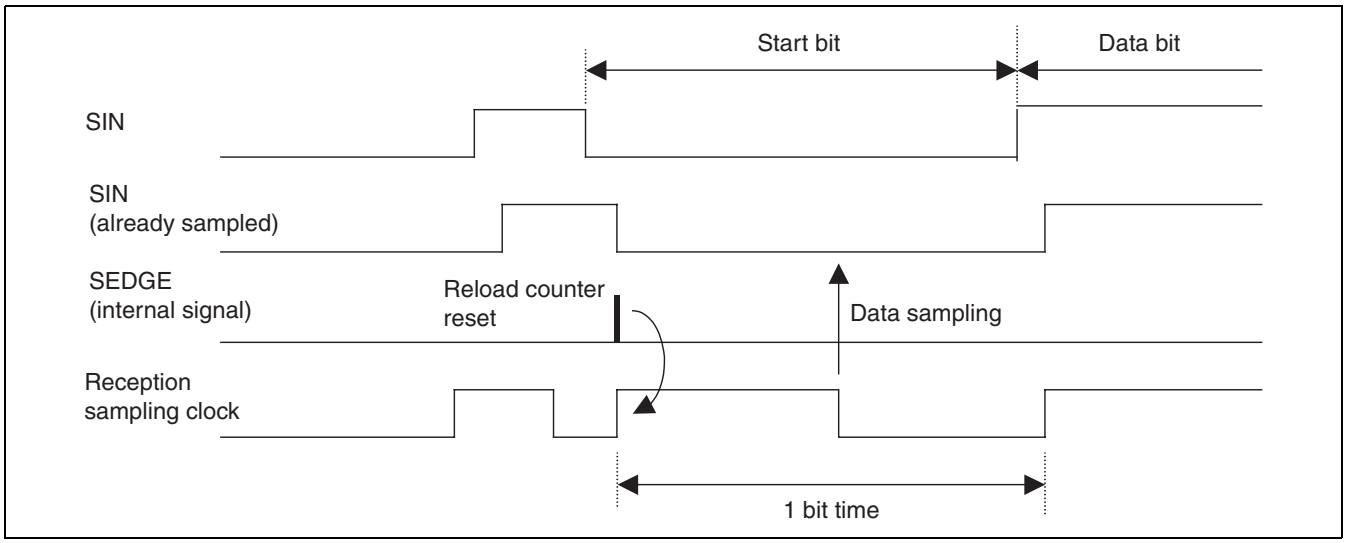
The data in the reception data register (RDR) will become valid, if no reception error occurs (SSR:PE, ORE, FRE = 0) when the reception data register full flag bit (SSR:RDRF) is set to "1".

● Clock selection

- The internal clock or external clock can be used.
- To use the external clock, set BGR:EXT to "1". In this case, the external clock is divided by the baud rate generator.

● Detection of the start bit

- In asynchronous mode, a start bit is identified by the falling edge of a SIN signal. Therefore, even when reception operation has been enabled (SCR:RXE = 1), such operation will not start unless the falling edge of a SIN signal is input.
- When the falling edge of a start bit is detected, the reception reload counter of the baud rate generator is reset and reloaded to start counting down. This allows sampling to be always performed using the central part of data.



● Stop bit

- 1 bit to 4 bits can be selected for the bit length.
- The reception data full flag bit (SSR:RDRF) is set to "1" when the first stop bit is detected.

● Detection of errors

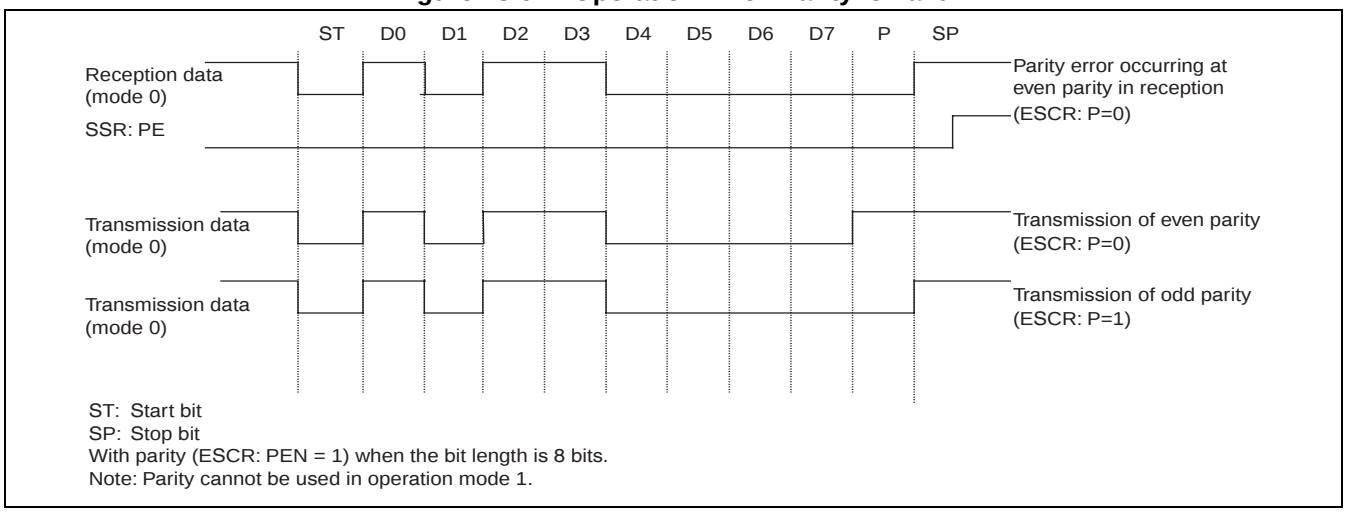
- In operation mode 0, parity errors, overrun errors and frame errors can be detected.
- In operation mode 1, overrun errors and frame errors can be detected. Parity errors, on the other hand, cannot be detected.

● Parity bit

- Addition of the parity bit can be selected only in operation mode 0. The parity enable bit (ESCR:PEN) can be used to determine the addition or omission of parity, while the parity selection bit (ESCR:P) can be used to select even parity or odd parity.
- Parity cannot be used in operation mode 1.

Figure 28.6-2 shows transmission/reception data when parity is valid.

Figure 28.6-2 Operation when Parity is Valid

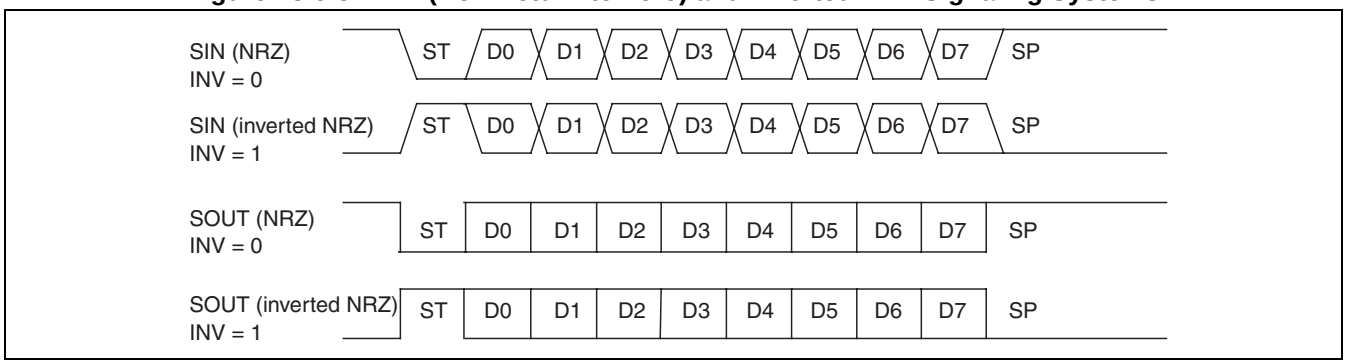


● Data signaling system

The NRZ (Non Return to Zero) signaling system (ESCR:INV = 0) or the inverted NRZ signaling system (ESCR:INV = 1) can be selected by setting the INV bit in the extended communication control register.

Figure 28.6-3 shows the NRZ and inverted NRZ signaling systems.

Figure 28.6-3 NRZ (Non Return to Zero) and Inverted NRZ Signaling Systems



● Data transfer system

LSB-first or MSB-first data bit transfer system can be selected.

28.7 Dedicated Baud Rate Generator

One of the following options can be selected for the transmission/reception clock source of the UART.

- Dedicated baud rate generator (reload counter)
- External clock input to the baud rate generator (reload counter)

■ UART Baud Rate Selection

One of the following two options can be selected for the baud rate.

● Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock, according to the set value.

To set the clock source, select the internal clock (BGR:EXT = 0).

● Baud rate achieved by dividing the external clock using the dedicated baud rate generator (reload counter)

The external clock is used as the clock source for the reload counter.

The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the external clock, according to the set value.

To set the clock source, select the external clock and the baud rate generator clock (BGR:EXT = 1).

This mode is made available on the assumption that an oscillator with a special frequency is divided for use.

<Notes>

- Set the external clock (EXT = 1) while the reload counter is stopped (BGR1/BGR0 = 15’h00).
- When the external clock has been selected (EXT = 1), the "H" and "L" widths of the external clock must be two or more peripheral clocks (PCLK).

28.7.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$V = \phi / b - 1$

V: Reload value
b: Baud rate
ϕ: Peripheral clock (PCLK), external clock frequency

(2)Example of calculation

If the peripheral clock (PCLK) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:

Reload value:

$V = (16 \times 1000000)/19200 - 1 = 832$

As a result, the baud rate is:

$b = (16 \times 1000000)/(832+1) = 19208$ bps

(3) Baud rate error

The following formula is used to calculate a baud rate error.

Error(%) = (calculated value - target value) / target value × 100

Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps

Reload value = $(20 \times 1000000)/153600 - 1 = 129$

Baud rate (calculated value) = $(20 \times 1000000)/(129+1) = 153846$ (bps)

Error (%) = $(153846 - 153600)/153600 \times 100 = 0.16$ (%)

<Notes>

- The reload counter halts when the reload value is set to "0".
- When the reload value is even-numbered, the "L" width of the reception serial clock is one peripheral clock (PCLK) cycle longer than the "H" width of the same serial clock. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
- Select 4 or a larger value for the reload value. However, data may not be able to be received properly, due to a baud rate error or reload settings.

■ Reload Values and Baud Rates for Different Peripheral Clock (PCLK) Frequencies

Table 28.7-1 Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
4M	-	-	-	-	-	0	4	0	5	0	7	0
2.5M	-	-	-	0	-	-	-	-	-	-	-	-
2M	-	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

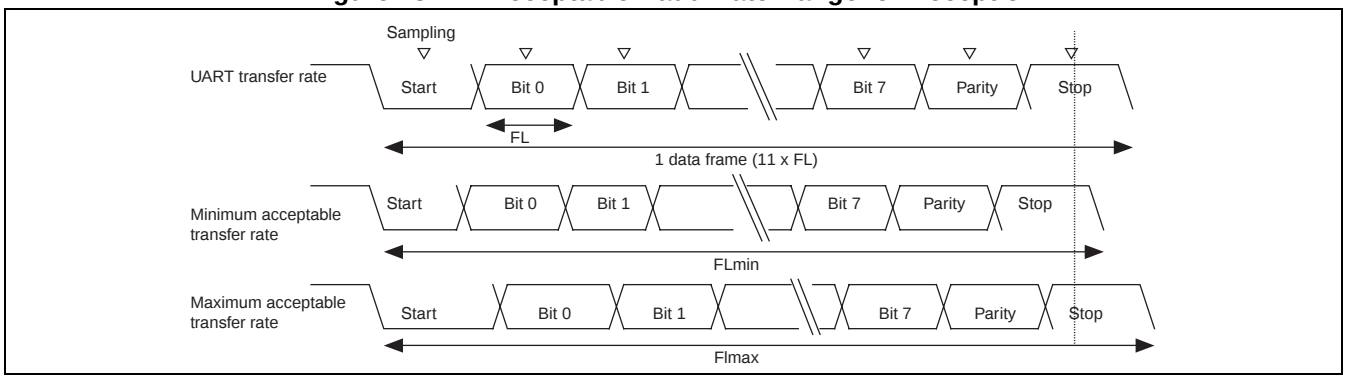
- Value: the value set in BGR1/BGR0 registers (decimal)
- ERR: baud rate error (%)

■ Acceptable Baud Rate Range for Reception

The following figure shows the range of acceptable baud rate differences at the transmission destination during reception.

The following calculation formula must be used to set a baud rate error for reception within the acceptable error range.

Figure 28.7-1 Acceptable Baud Rate Range for Reception



As shown in the figure, the sampling timing for reception data is determined by the counter selected by the BGR1/BGR0 registers after a start bit is detected. If all data including the last data (stop bit) can fit in this sampling timing, reception can be performed successfully.

In theory, the following is expected when this is applied to 11-bit reception.

When the sampling timing margin is equivalent of two clocks of the peripheral clock (PCLK) (ϕ), the minimum acceptable transfer rate (FLmin) is as follows:

FLmin = (11 bits $\times$ (V + 1) - (V + 1)/2 + 2)/ϕ = (21V + 25)/2ϕ (s)
V: reload value ϕ: peripheral clock (PCLK)

Consequently, the maximum receivable baud rate at the transmission destination (BGmax) is as follows:

BGmax = 11/FLmin = 22ϕ/(21V+25) (bps)
V: reload value ϕ: peripheral clock (PCLK)

Likewise, the maximum acceptable transfer rate (Flmax) can be calculated as shown below:

FLmax = (11 bits $\times$ (V + 1) + (V + 1)/2 - 2)/ϕ = (23V + 19)/2ϕ (s)
V: reload value ϕ: peripheral clock (PCLK)

Consequently, the minimum receivable baud rate at the transmission destination (BGmin) is as follows:

BGmin = 11/FLmax = 22ϕ/(23V+19) (bps)
V: reload value ϕ: peripheral clock (PCLK)

Based on the aforementioned calculation formulas for the minimum/maximum baud rates, the acceptable baud rate error between the UART and transmission destination can be calculated as shown below.

Table 28.7-2 Acceptable Baud Rate Error

Reload value (V)	Maximum acceptable baud rate error	Minimum acceptable baud rate error
3	0%	0
10	+2.98%	-2.81%
50	+4.37%	-4.02%
100	+4.56%	-4.18%
200	+4.66%	-4.26%
32767	+4.76%	-4.35%

<Note>

The accuracy of reception depends on the number of bits per frame, the peripheral clock (PCLK) and the reload value. The accuracy becomes higher as the peripheral clock (PCLK) and the division ratio become higher.

■ External Clock

The baud rate generator divides the external clock, when "1" is written to the EXT bit in the baud rate generator register 1, 0 (BGR1, BGR0).

<Note>

The UART synchronizes external clock signals with the internal clock. Therefore, the operation becomes unstable when an external clock which cannot be synchronized is used.

■ Functions of Reload Counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the external or internal clock.

■ Starting a Count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

■ Restart

The reload counter restarts under the following conditions.

- For both transmission and reception reload counters
  Programmable reset (SCR:UPCL bit)
- For reception reload counter
  Detecting the falling edge of a start bit in asynchronous mode

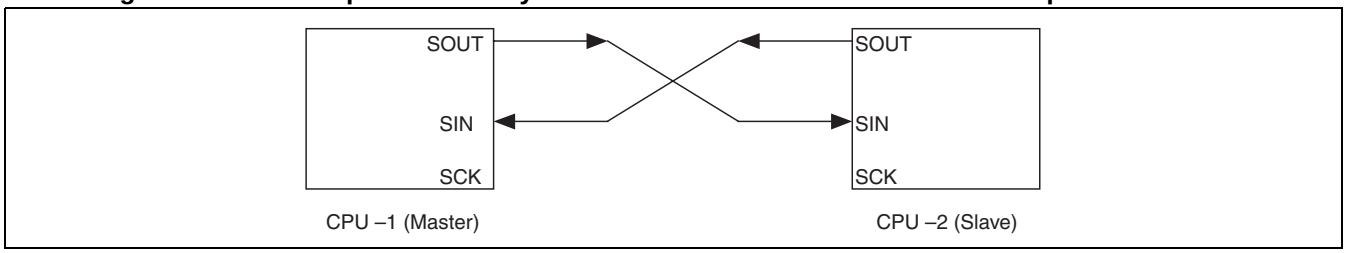
28.8 Setup Procedure and Program Flow for Operation Mode 0 (Asynchronous Normal Mode)

Asynchronous serial two-way communication is enabled in operation mode 0.

■ Connection between CPUs

Two-way communication should be selected for operation mode 0 (normal mode). Two CPUs are connected to each other, as shown in Figure 28.8-1.

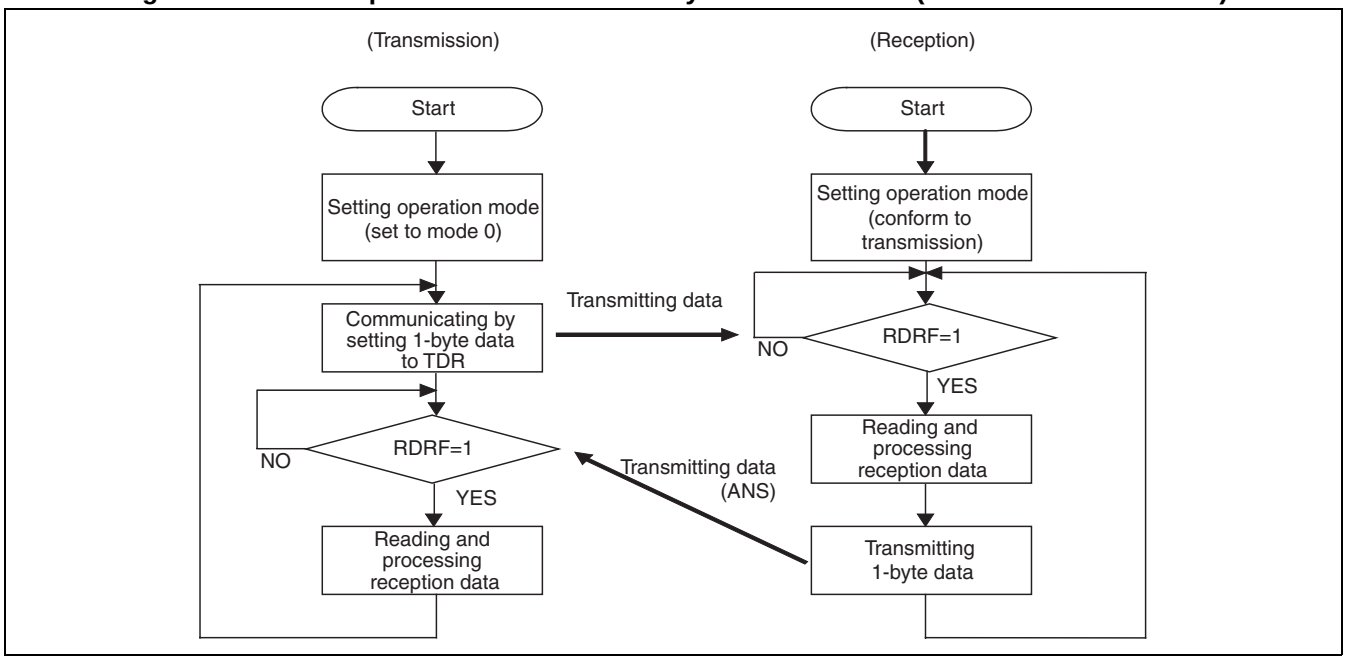
Figure 28.8-1 Example of Two-way Communication Connection for UART Operation Mode 0



■ Flowchart

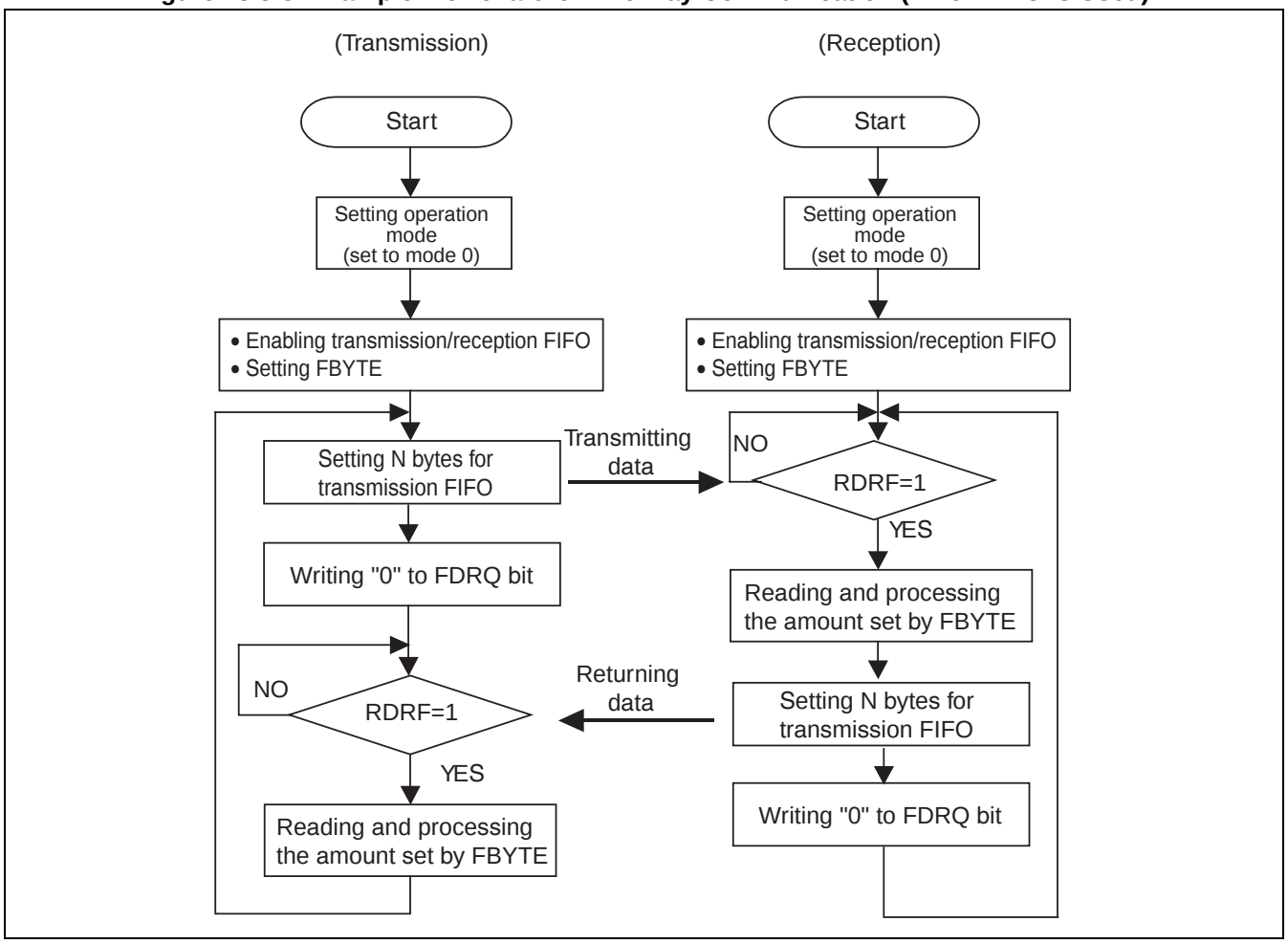
● When FIFO is not used

Figure 28.8-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



● When FIFO is used

Figure 28.8-3 Example Flowchart for Two-way Communication (When FIFO is Used)



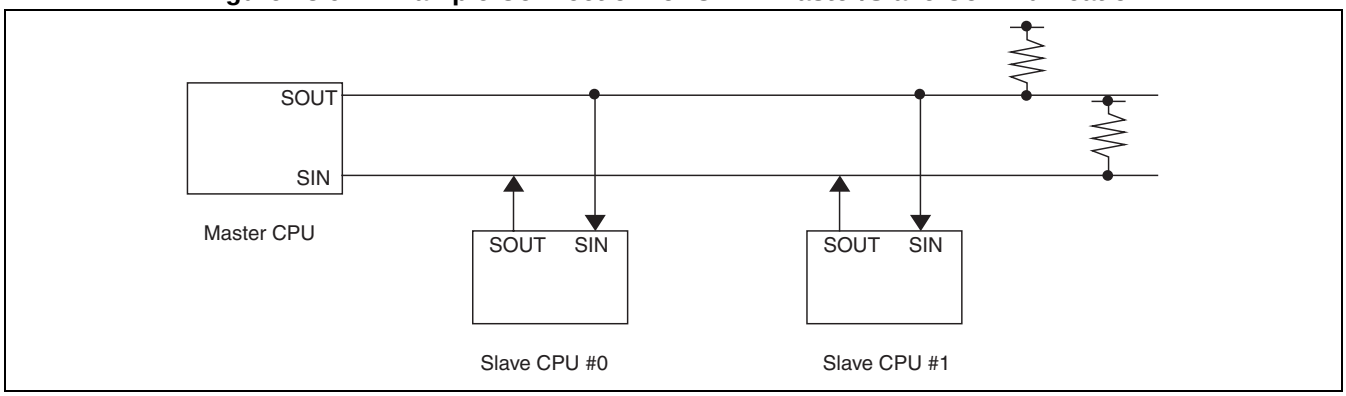
28.9 Setup Procedure and Program Flow for Operation Mode 1 (Asynchronous Multi-processor Mode)

In operation mode 1 (multi-processor mode), communication among multiple CPUs is enabled through master/slave connection. The connected CPUs can be used as a master/slave.

■ Connection among CPUs

In this master/slave communication, one master CPU and more than one slave CPU are connected to two common communication lines, as shown in Figure 28.9-1, to configure a communication system. The UART can be used by both the master and slaves.

Figure 28.9-1 Example Connection for UART Master/Slave Communication



■ Function Selection

For master/slave communication, select the operation mode and data transfer system shown in Table 28.9-1.

Table 28.9-1 Selection of Master/Slave Communication Function

	Operation mode		Data	Parity	Stop bit	Bit direction
	Master CPU	Slave CPU				
Address transmission/ reception	Mode 1 (AD bit transmission)	Mode 1 (AD bit reception)	AD = 1 + 7-bit or 8-bit address	None	1 bit to 4 bits	LSB first or MSB first
Data transmission/ reception			AD = 0 + 7-bit or 8-bit data			

<Note>

Use half word access for transmission/reception data (RDR/TDR) in operation mode 1.

● Communication procedure

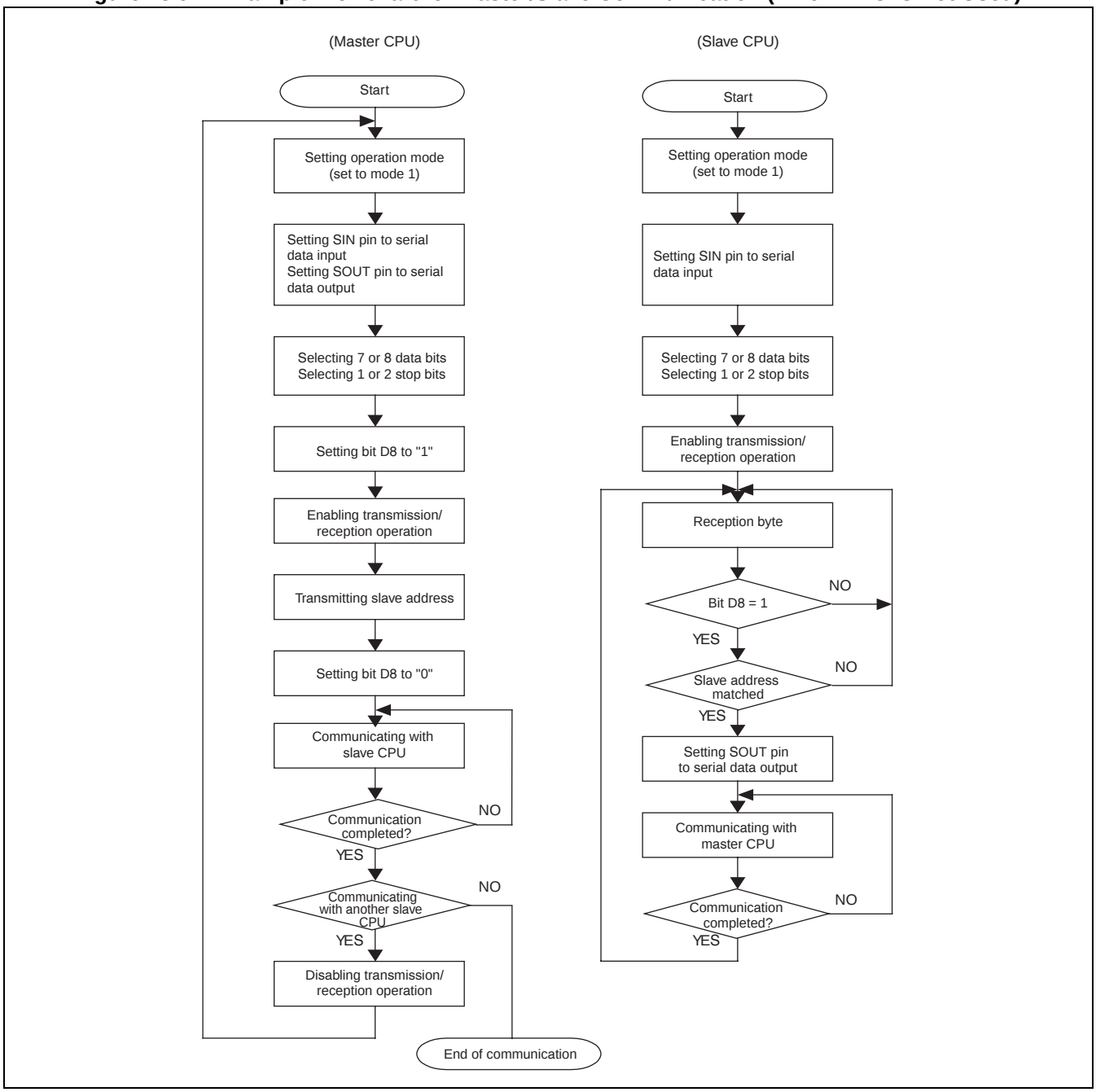
Communication is started when the master CPU transmits address data, where bit D8 is treated as "1". This data is used to select a slave CPU which will be the communication destination. Each slave CPU judges the address data on a program, and communicates (normal data) with the master CPU when the data matches its assigned address.

Figure 28.9-2 and Figure 28.9-3 show flowcharts for the master/slave communication (multi-processor mode).

■ Flowcharts

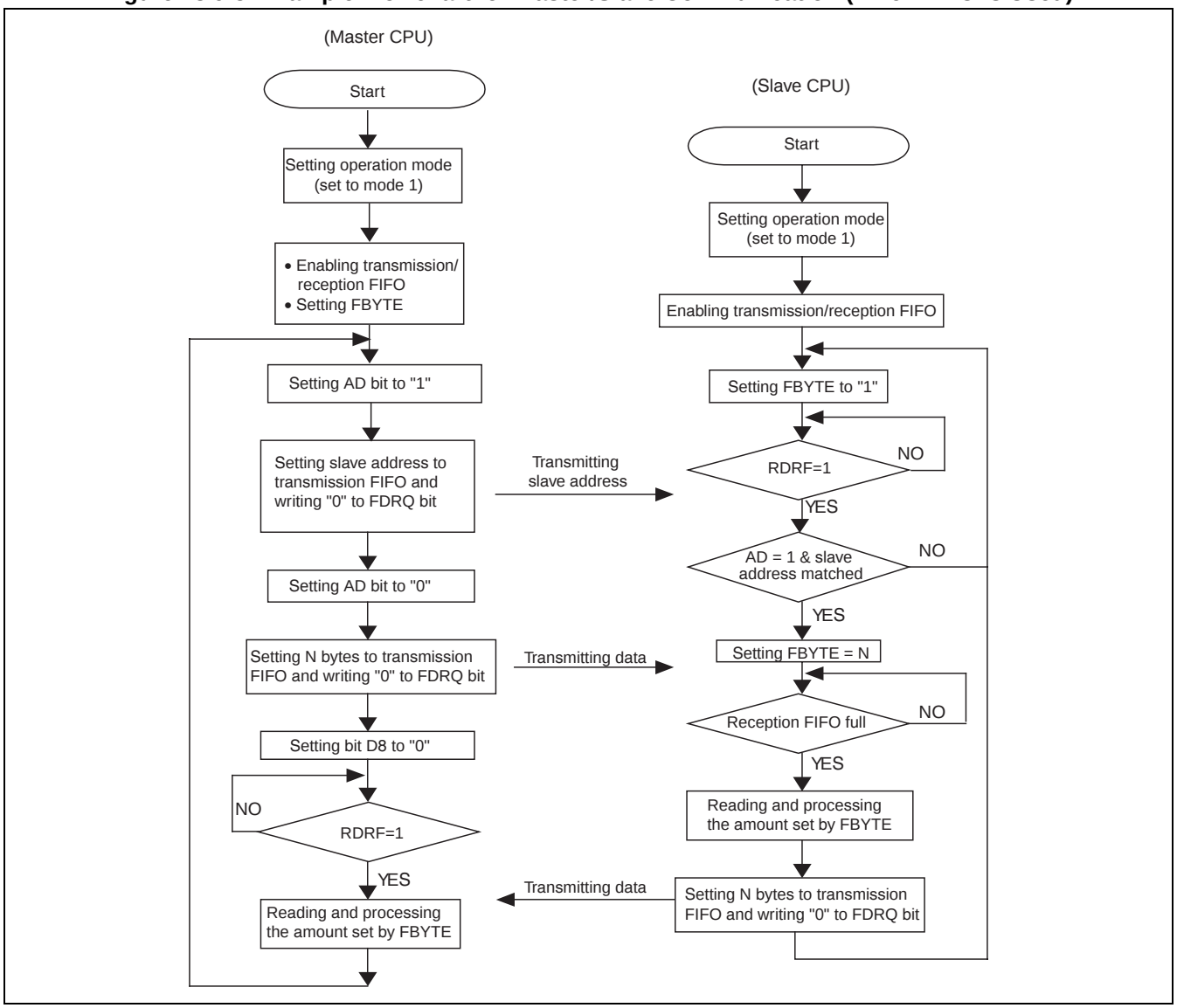
● When FIFO is not used:

Figure 28.9-2 Example Flowchart for Master/Slave Communication (When FIFO is Not Used)



● When FIFO is used

Figure 28.9-3 Example Flowchart for Master/Slave Communication (When FIFO is Used)



28.10 CSIO (Clock Synchronous Serial Interface)

Among all the functions of the multi-function serial interface, this section describes the CSIO functions that are supported in operation mode 2.

- CSIO (Clock Synchronous Serial Interface)
- Overview of CSIO (Clock Synchronous Serial Interface)
- Registers of CSIO (Clock Synchronous Serial Interface)
  - Serial Control Register (SCR)
  - Serial Mode Register (SMR)
  - Serial Status Register (SSR)
  - Extended Serial Control Register (ESCR)
  - Reception Data Register / Transmission Data Register (RDR/TDR)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
  - Serial mode selection registers (SSEL0123, SSEL4567)
- Reception data mirror registers/ transmission data mirror registers (RDRM/TDRM)
- Interrupts of CSIO (Clock Synchronous Serial Interface)
  - Occurrence of Reception Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing
  - Occurrence of Transmission Interrupts and Flag Set Timing
  - Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing
- Operation of CSIO (Clock Synchronous Serial Interface)
- Dedicated Baud Rate Generator
  - Setting Baud Rate
- Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

28.11 Overview of CSIO (Clock Synchronous Serial Interface)

CSIO (Clock Synchronous Serial Interface) is a general-purpose interface for serial data communication, which allows synchronous communications with external units (SPI supported). In addition, this interface comes with transmission/reception FIFO (up to 16 bytes each).

■ Functions of CSIO (Clock Synchronous Serial Interface)

		Function
1	Data buffer	• Full-duplex double buffer (when FIFO is not used) • Transmission/reception FIFO (up to 16 bytes each) (when FIFO is used)*
2	Transfer system	• Clock synchronization (no start bit / no stop bit) • Master/slave function • SPI supported (both master & slaves supported)
3	Baud rate	• Dedicated baud rate generator available (15-bit reload counter configuration, in master operation) • External clock can be input (in slave operation)
4	Data length	Variable from 5 bits to 9 bits
5	Reception error detection	Overrun error
6	Interrupt request	• Reception interrupt (completion of reception, overrun error) • Transmission interrupt (transmission data empty, transmission bus idle) • Transmission FIFO interrupt (when transmission FIFO is empty) • DMA transfer support function for transmission and reception
7	Synchronous mode	Master or slave function
8	Pin access	Serial data output pin can be set to "H".
9	4-channel simultaneous communication	4-channel simultaneous communication is available for ch.0 to ch.3 and ch.4 to ch.7.
10	FIFO options	• Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)* • Transmission FIFO or reception FIFO selectable • Transmission data can be resent. • The interrupt timing for reception FIFO can be modified by software. • FIFO reset is supported separately.

*: There is no FIFO between ch.0 and ch.7

28.12 Registers of CSIO (Clock Synchronous Serial Interface)

This section lists the registers of CSIO (clock synchronous serial interface).

■ List of Registers of CSIO (Clock Synchronous Serial Interface)

Table 28.12-1 List of Registers of CSIO (Clock Synchronous Serial Interface) (1 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
Common to 0 to 3	SSEL0123	Serial mode select register 0123	28.12.10
Common to 4 to 7	SSEL4567	Serial mode select register 4567	28.12.10
0	SCR0	Serial control register 0	28.12.1
	SMR0	Serial mode register 0	28.12.2
	ESCR0	Extended serial control register 0	28.12.4
	BGR0	Baud rate generator register 0	28.12.6
	SSR0	Serial status register 0	28.12.3
	RDR0	Received data register 0	28.12.5
	TDR0	Transmitted data register 0	28.12.5
	RDRM0	Received data mirror register 0	28.12.11
	TDRM0	Transmitted data mirror register 0	28.12.11
1	SCR1	Serial control register 1	28.12.1
	SMR1	Serial mode register 1	28.12.2
	ESCR1	Extended serial control register 1	28.12.4
	BGR1	Baud rate generator register 1	28.12.6
	SSR1	Serial status register 1	28.12.3
	RDR1	Received data register 1	28.12.5
	TDR1	Transmitted data register 1	28.12.5
	RDRM1	Received data mirror register 1	28.12.11
	TDRM1	Transmitted data mirror register 1	28.12.11

Table 28.12-1 List of Registers of CSIO (Clock Synchronous Serial Interface) (2 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
2	SCR2	Serial control register 2	28.12.1
	SMR2	Serial mode register 2	28.12.2
	ESCR2	Extended serial control register 2	28.12.4
	BGR2	Baud rate generator register 2	28.12.6
	SSR2	Serial status register 2	28.12.3
	RDR2	Received data register 2	28.12.5
	TDR2	Transmitted data register 2	28.12.5
	RDRM2	Received data mirror register 2	28.12.11
	TDRM2	Transmitted data mirror register 2	28.12.11
3	SCR3	Serial control register 3	28.12.1
	SMR3	Serial mode register 3	28.12.2
	ESCR3	Extended serial control register 3	28.12.4
	BGR3	Baud rate generator register 3	28.12.6
	SSR3	Serial status register 3	28.12.3
	RDR3	Received data register 3	28.12.5
	TDR3	Transmitted data register 3	28.12.5
	RDRM3	Received data mirror register 3	28.12.11
	TDRM3	Transmitted data mirror register 3	28.12.11
4	SCR4	Serial control register 4	28.12.1
	SMR4	Serial mode register 4	28.12.2
	ESCR4	Extended serial control register 4	28.12.4
	BGR4	Baud rate generator register 4	28.12.6
	SSR4	Serial status register 4	28.12.3
	RDR4	Received data register 4	28.12.5
	TDR4	Transmitted data register 4	28.12.5
	RDRM4	Received data mirror register 4	28.12.11
	TDRM4	Transmitted data mirror register 4	28.12.11

Table 28.12-1 List of Registers of CSIO (Clock Synchronous Serial Interface) (3 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
5	SCR5	Serial control register 5	28.12.1
	SMR5	Serial mode register 5	28.12.2
	ESCR5	Extended serial control register 5	28.12.4
	BGR5	Baud rate generator register 5	28.12.6
	SSR5	Serial status register 5	28.12.3
	RDR5	Received data register 5	28.12.5
	TDR5	Transmitted data register 5	28.12.5
	RDRM5	Received data mirror register 5	28.12.11
	TDRM5	Transmitted data mirror register 5	28.12.11
6	SCR6	Serial control register 6	28.12.1
	SMR6	Serial mode register 6	28.12.2
	ESCR6	Extended serial control register 6	28.12.4
	BGR6	Baud rate generator register 6	28.12.6
	SSR6	Serial status register 6	28.12.3
	RDR6	Received data register 6	28.12.5
	TDR6	Transmitted data register 6	28.12.5
	RDRM6	Received data mirror register 6	28.12.11
	TDRM6	Transmitted data mirror register 6	28.12.11
7	SCR7	Serial control register 7	28.12.1
	SMR7	Serial mode register 7	28.12.2
	ESCR7	Extended serial control register 7	28.12.4
	BGR7	Baud rate generator register 7	28.12.6
	SSR7	Serial status register 7	28.12.3
	RDR7	Received data register 7	28.12.5
	TDR7	Transmitted data register 7	28.12.5
	RDRM7	Received data mirror register 7	28.12.11
	TDRM7	Transmitted data mirror register 7	28.12.11

Table 28.12-1 List of Registers of CSIO (Clock Synchronous Serial Interface) (4 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
8	SCR8	Serial control register 8	28.12.1
	SMR8	Serial mode register 8	28.12.2
	ESCR8	Extended serial control register 8	28.12.4
	BGR8	Baud rate generator register 8	28.12.6
	SSR8	Serial status register 8	28.12.3
	RDR8	Received data register 8	28.12.5
	TDR8	Transmitted data register 8	28.12.5
	FCR18	FIFO control register 18	28.12.7
	FCR08	FIFO control register 08	28.12.8
	FBYTE18	FIFO1 byte register 8	28.12.9
	FBYTE28	FIFO2 byte register 8	28.12.9
9	SCR9	Serial control register 9	28.12.1
	SMR9	Serial mode register 9	28.12.2
	ESCR9	Extended serial control register 9	28.12.4
	BGR9	Baud rate generator register 9	28.12.6
	SSR9	Serial status register 9	28.12.3
	RDR9	Received data register 9	28.12.5
	TDR9	Transmitted data register 9	28.12.5
	FCR19	FIFO control register 19	28.12.7
	FCR09	FIFO control register 09	28.12.8
	FBYTE19	FIFO1 byte register 9	28.12.9
	FBYTE29	FIFO2 byte register 9	28.12.9

Table 28.12-1 List of Registers of CSIO (Clock Synchronous Serial Interface) (5 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
10	SCR10	Serial control register 10	28.12.1
	SMR10	Serial mode register 10	28.12.2
	ESCR10	Extended serial control register 10	28.12.4
	BGR10	Baud rate generator register 10	28.12.6
	SSR10	Serial status register 10	28.12.3
	RDR10	Received data register 10	28.12.5
	TDR10	Transmitted data register 10	28.12.5
	FCR110	FIFO control register 110	28.12.7
	FCR010	FIFO control register 010	28.12.8
	FBYTE110	FIFO1 byte register 10	28.12.9
	FBYTE210	FIFO2 byte register 10	28.12.9
11	SCR11	Serial control register 11	28.12.1
	SMR11	Serial mode register 11	28.12.2
	ESCR11	Extended serial control register 11	28.12.4
	BGR11	Baud rate generator register 11	28.12.6
	SSR11	Serial status register 11	28.12.3
	RDR11	Received data register 11	28.12.5
	TDR11	Transmitted data register 11	28.12.5
	FCR111	FIFO control register 111	28.12.7
	FCR011	FIFO control register 011	28.12.8
	FBYTE111	FIFO1 byte register 11	28.12.9
	FBYTE211	FIFO2 byte register 11	28.12.9

Table 28.12-2 Bit Assignment of CSIO (Clock Synchronous Serial Interface)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
-	-								-							
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

28.12.1 Serial Control Register (SCR)

The serial control register (SCR) enables/disables transmission/reception interrupts, transmission idle interrupts and transmission/reception operations. This register can also set SPI connection and reset CSIO.

■ Serial Control Register (SCR)

Figure 28.12-1 shows the bit structure of the serial control register (SCR), and Table 28.12-3 describes the function of each bit.

Figure 28.12-1 Bit Structure of Serial Control Register (SCR)

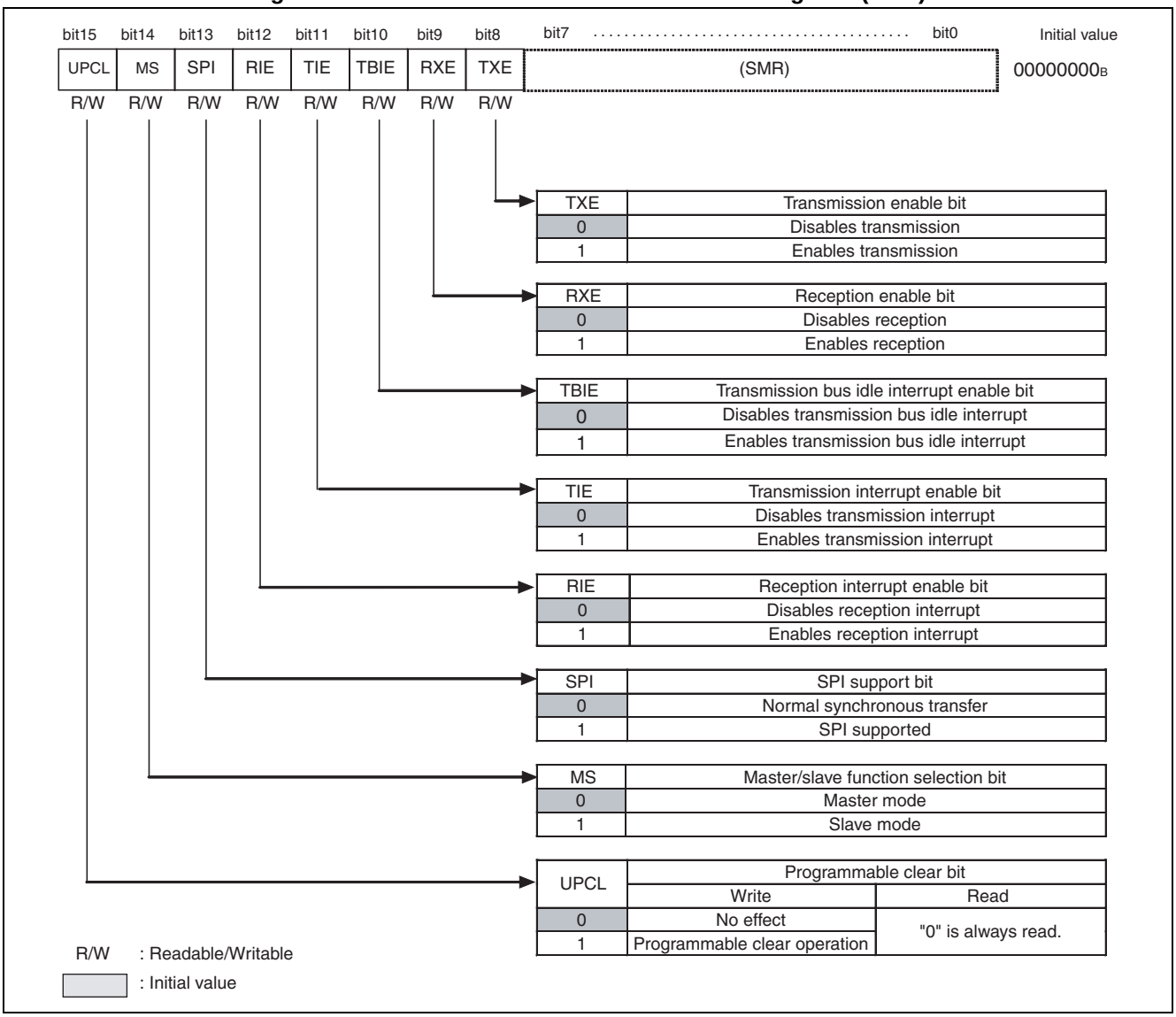

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
UPCL MS SPI RIE TIE TBIE RXE TXE (SMR) 00000000B
R/W R/W R/W R/W R/W R/W R/W R/W
TXE Transmission enable bit
0 Disables transmission
1 Enables transmission
RXE Reception enable bit
0 Disables reception
1 Enables reception
TBIE Transmission bus idle interrupt enable bit
0 Disables transmission bus idle interrupt
1 Enables transmission bus idle interrupt
TIE Transmission interrupt enable bit
0 Disables transmission interrupt
1 Enables transmission interrupt
RIE Reception interrupt enable bit
0 Disables reception interrupt
1 Enables reception interrupt
SPI SPI support bit
0 Normal synchronous transfer
1 SPI supported
MS Master/slave function selection bit
0 Master mode
1 Slave mode
UPCL Programmable clear bit
Write Read
0 No effect "0" is always read.
1 Programmable clear operation
R/W : Readable/Writable
: Initial value

Table 28.12-3 Functional Description of Each Bit of Serial Control Register (SCR) (1 / 2)

Bit name		Function
bit15	UPCL: Programmable clear bit	This bit is used to initialize the internal state of the CSIO. Setting the bit to "1": • The CSIO will be reset directly (software reset). The register setting, however, will be retained. In this case, communication of the data which is being transmitted or received will be cut off immediately. • The baud rate generator will reload the value set in BGR1/BGR0 registers, and then restart the operation. • All the transmission/reception interrupt sources (TDRE, TBI, RDRF and ORE) will be initialized ("1100_B"). • Setting the bit to "0": No effect on the operation • Reading this bit always returns "0". Note: Execute the programmable clear operation after disabling interrupts. Execute the programmable clear operation after disabling FIFO (FE2, FE1 = 0) when FIFO is used.
bit14	MS: Master/slave function selection bit	This bit is used to select master or slave mode. Setting the bit to "0" selects master mode. Setting the bit to "1" selects slave mode. Note: The external clock will be input directly, if SMR:SCKE is set to "0" when slave mode is selected.
bit13	SPI: SPI support bit	This bit is used to enable communication supporting SPI. Setting the bit to "0" enables normal synchronous communication. Setting the bit to "1" enables SPI support.
bit12	RIE: Reception interrupt enable bit	• This bit is used to enable/disable the output of reception interrupt requests to the CPU. • A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or when any of the error flag bits (ORE) is set to "1".
bit11	TIE: Transmission interrupt enable bit	• This bit is used to enable/disable the output of transmission interrupt requests to the CPU. • A transmission interrupt request is output when the TIE and TDRE bits are set to "1".
bit10	TBIE: Transmission bus idle interrupt enable bit	• This bit is used to enable/disable the output of transmission bus idle interrupt requests to the CPU. • A transmission bus idle interrupt request is output when the TBIE and TBI bits are set to "1".
bit9	RXE: Reception enable bit	This bit is used to enable/disable CSIO reception operation. Setting the bit to "0" disables data frame reception operation. Setting the bit to "1" enables data frame reception operation. Note: If the reception operation is disabled (RXE = 0) during the reception, the operation will be terminated immediately.

Table 28.12-3 Functional Description of Each Bit of Serial Control Register (SCR) (2 / 2)

Bit name		Function
bit8	TXE: Transmission enable bit	This bit is used to enable/disable CSIO transmission operation. Setting the bit to "0" disables data frame transmission operation. Setting the bit to "1" enables data frame transmission operation. Note: If the transmission operation is disabled (TXE = 0) during the transmission, the operation will be terminated immediately.

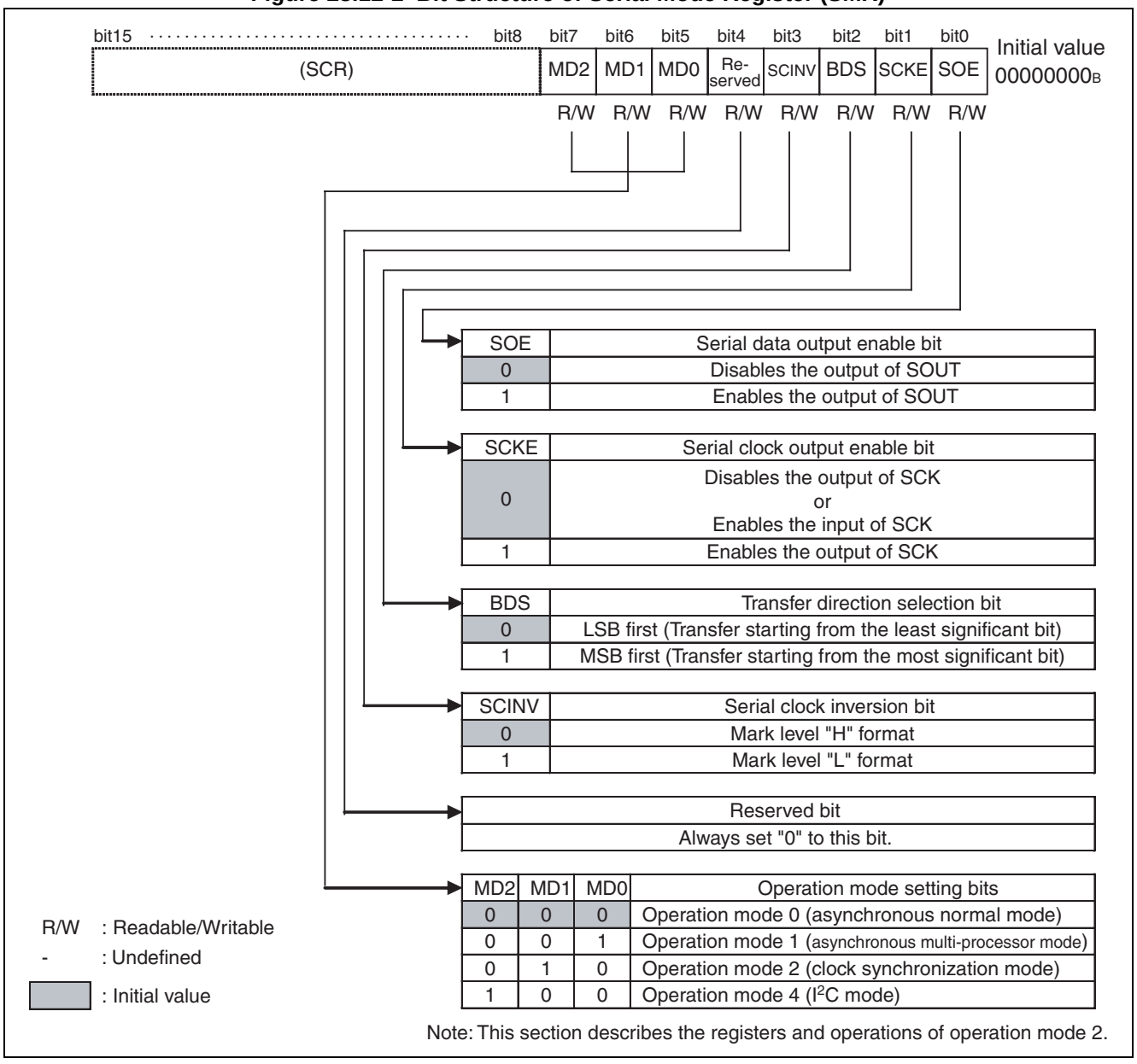
28.12.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, selects the transfer direction and serial clock inversion, and enables or disables the output to the serial data and serial clock pins.

■ Serial Mode Register (SMR)

Figure 28.12-2 shows the bit structure of the serial mode register (SMR), and Table 28.12-4 describes the function of each bit.

Figure 28.12-2 Bit Structure of Serial Mode Register (SMR)


bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0
(SCR) MD2 MD1 MD0 Re-served SCINV BDS SCKE SOE
Initial value 00000000B
R/W R/W R/W R/W R/W R/W R/W R/W
SOE: Serial data output enable bit
0: Disables the output of SOUT
1: Enables the output of SOUT
SCKE: Serial clock output enable bit
0: Disables the output of SCK or Enables the input of SCK
1: Enables the output of SCK
BDS: Transfer direction selection bit
0: LSB first (Transfer starting from the least significant bit)
1: MSB first (Transfer starting from the most significant bit)
SCINV: Serial clock inversion bit
0: Mark level "H" format
1: Mark level "L" format
Reserved bit
Always set "0" to this bit.
MD2 MD1 MD0: Operation mode setting bits
0 0 0: Operation mode 0 (asynchronous normal mode)
0 0 1: Operation mode 1 (asynchronous multi-processor mode)
0 1 0: Operation mode 2 (clock synchronization mode)
1 0 0: Operation mode 4 (I2C mode)
R/W : Readable/Writable
- : Undefined
: Initial value

Note: This section describes the registers and operations of operation mode 2.

Table 28.12-4 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	These bits are used to select the operation mode. "000_B": Selects operation mode 0 (asynchronous normal mode) "001_B": Selects operation mode 1 (asynchronous multi-processor mode) "010_B": Selects operation mode 2 (clock synchronization mode) "100_B": Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 2 (clock synchronization mode). Note: Settings other than above are prohibited. To switch the operation mode, execute the programmable clear operation first (SCR:UPCL = 1). And then, after setting the operation mode, set each register.
bit4	Reserved bit	Always set "0" to this bit.
bit3	SCINV: Serial clock inversion bit	This bit is used to invert the serial clock format. Setting the bit to "0": • Changes the mark level of the serial clock output to "H". • Transmission data is output, being synchronized with the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. • Reception data is sampled at the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. Setting the bit to "1": • Changes the mark level of the serial clock output to "L". • Transmission data is output, being synchronized with the rising edge (normal transfer) or the falling edge (SPI transfer) of the serial clock. • Reception data is sampled at the falling edge (normal transfer) or the rising edge (SPI transfer) of the serial clock. Note: Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit2	BDS: Transfer direction selection bit	This bit is used to determine the transfer priority for transfer serial data: whether the least significant bit should be transferred first (LSB first, BDS = 0) or the most significant bit should be transferred first (MSB first, BDS = 1). Note: Set this bit when transmission and reception are disabled (TXE = RXE = 0).
bit1	SCKE: Serial clock output enable bit	This bit is used to control the I/O port of the serial clock. Setting the bit to "0": The output of SCK "H" or the input of SCK will be enabled. To use it as a SCK input, set a general-purpose I/O port as the input port. Setting the bit to "1" enables the output of SCK.
bit0	SOE: Serial data output enable bit	This bit is used to enable/disable the output of serial data. Setting the bit to "0" enables the output of the "H" level of SOUT. Setting the bit to "1" enables the output of SOUT.

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when SCR and SMR are written simultaneously with 16-bit write access, SCR reflects the written content.

28.12.3 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status, and also checks and clears the reception error flag.

■ Serial Status Register (SSR)

Figure 28.12-3 shows the bit structure of the serial status register (SSR) and Table 28.12-5 describes the function of each bit.

Figure 28.12-3 Bit Structure of Serial Status Register (SSR)

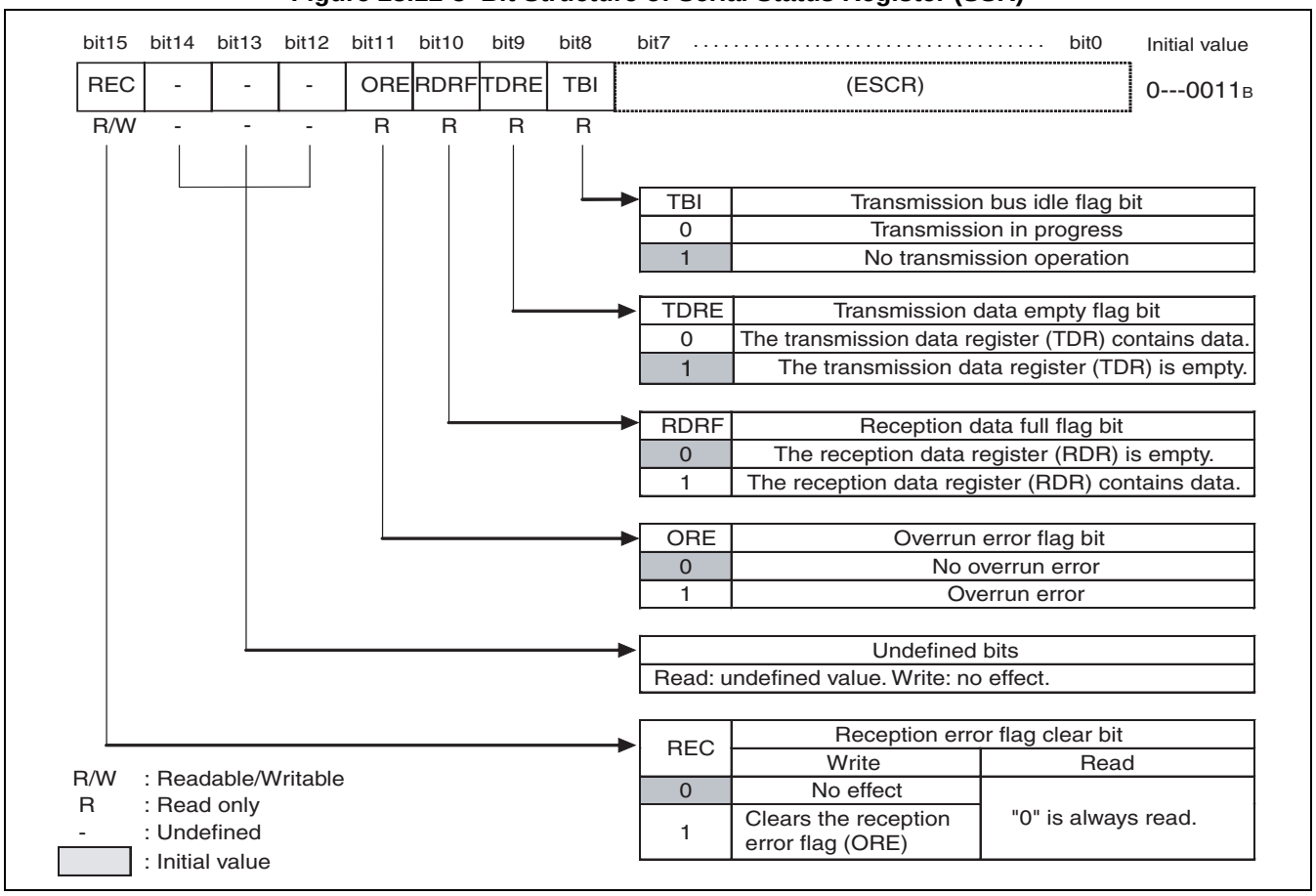

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
REC - - - ORE RDRF TDRE TBI (ESCR) 0---0011B
R/W - - - R R R R
TBI: 0 Transmission in progress; 1 No transmission operation
TDRE: 0 The transmission data register (TDR) contains data.; 1 The transmission data register (TDR) is empty.
RDRF: 0 The reception data register (RDR) is empty.; 1 The reception data register (RDR) contains data.
ORE: 0 No overrun error; 1 Overrun error
Undefined bits
Read: undefined value. Write: no effect.
REC: 0 No effect; 1 Clears the reception error flag (ORE)
Read: "0" is always read.
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.12-5 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the ORE flag in the serial status register (SSR). • Writing "1" clears the error flag. • Writing "0" has no effect. Reading this bit always returns "0".
bit14 to bit12	Undefined bits	Read: undefined value Write: no effect

Table 28.12-5 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit11	ORE: Overrun error flag bit	• This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of the reception FIFO, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	• This flag indicates the status of the reception data register (RDR). • The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. • A reception interrupt request is output when the RDRF and RIE bits are set to "1". • RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. • During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO. If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. • This bit is cleared to "0" when the reception FIFO, if used, becomes empty.
bit9	TDRE: Transmission data empty flag bit	• This flag indicates the status of the transmission data register (TDR). • When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. • A transmission interrupt request is output when the TDRE and TIE bits are set to "1". • The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • For information about the set/reset timings of the TDRE bit for when the transmission FIFO is used, refer to Section "28.13.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".
bit8	TBI: Transmission bus idle flag bit	• This bit indicates that the CSIO is not performing transmission operation. • The bit is set to "0" when data is written to the transmission data register (TDR). • The bit is set to "1" when the transmission data register (TDR) is empty (TDRE = 1) and no transmission operation is in progress. • The TDRE bit becomes "1" when the UPCL bit in the serial control register (SCR) is set to "1". • A transmission interrupt request is output when this bit is "1" and a transmission bus idle interrupt is enabled (SCR:TBIE = 1).

28.12.4 Extended Serial Control Register (ESCR)

The extended serial control register (ESCR) can be used to set the transmission/reception data length, select data transmission/reception wait, and fix the serial output to "H".

■ Bit Structure of the Extended Serial Control Register (ESCR)

Figure 28.12-4 shows the bit structure of the extended serial control register (ESCR) and Table 28.12-6 describes the function of each bit.

Figure 28.12-4 Bit Structure of Extended Serial Control Register (ESCR)

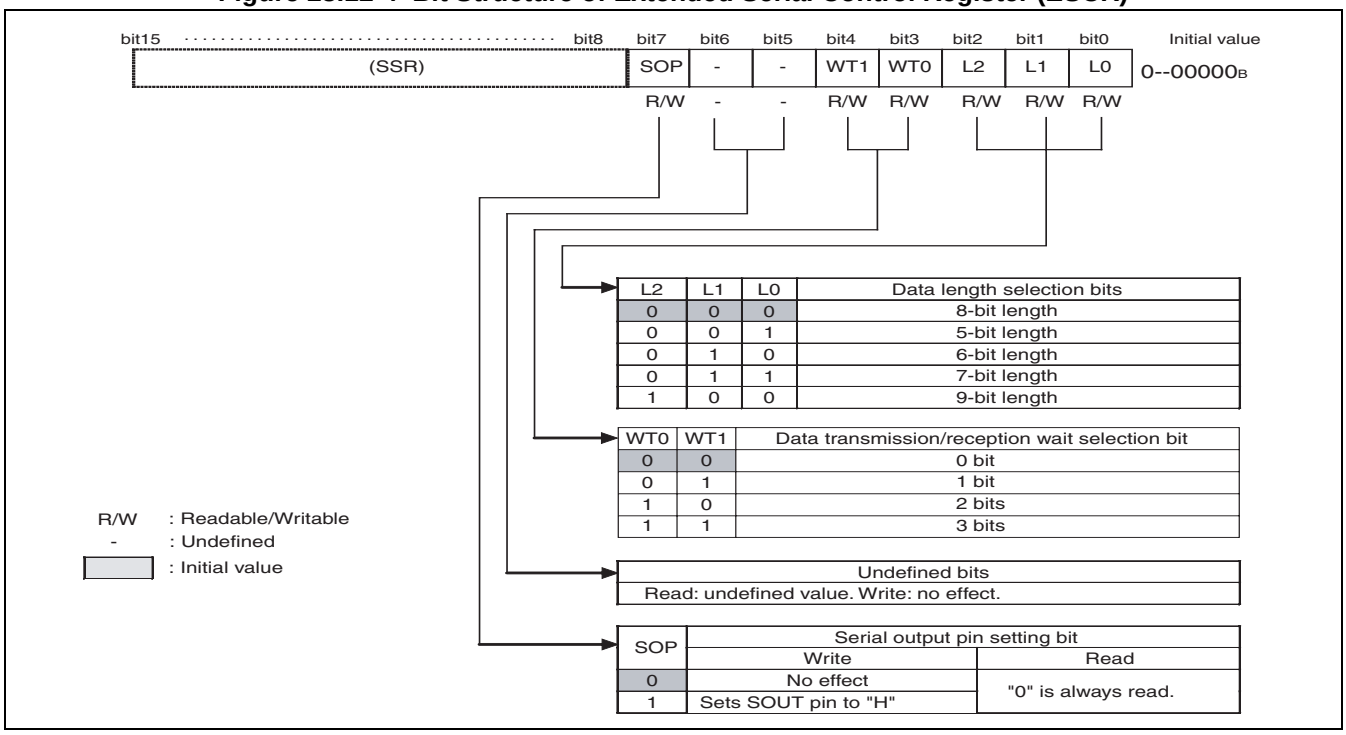

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(SSR) SOP - - WT1 WT0 L2 L1 L0 0--00000B
R/W - - R/W R/W R/W R/W R/W
L2 L1 L0 Data length selection bits
0 0 0 8-bit length
0 0 1 5-bit length
0 1 0 6-bit length
0 1 1 7-bit length
1 0 0 9-bit length
WT0 WT1 Data transmission/reception wait selection bit
0 0 0 bit
0 1 1 bit
1 0 2 bits
1 1 3 bits
Undefined bits
Read: undefined value. Write: no effect.
SOP Serial output pin setting bit
Write Read
0 No effect "0" is always read.
1 Sets SOUT pin to "H"
R/W : Readable/Writable
- : Undefined
: Initial value

Table 28.12-6 Functional Description of Each Bit of Extended Serial Control Register (ESCR)

Bit name		Function
bit7	SOP: Serial output pin setting bit	• This bit is used to set the serial output pin to "H". The SOUT pin is set to "H" when "1" is written to this bit. It is not necessary to write "0" to this bit after that. • Reading this bit always returns "0". Note: Do not set this bit during serial data transmission.
bit6, bit5	Undefined bits	Read: undefined value Write: no effect
bit4, bit3	WT1,WT0: Data transmission/ reception wait selection bit	In master mode, the wait number is specified to transmission/reception of sequential data. In slave mode, it will be "00" operation. • If "00_B" is set, SCK is output continuously. • If "01_B" is set, SCK is output after 1 bit time wait. • If "10_B" is set, SCK is output after 2 bits time wait. • If "11_B" is set, SCK is output after 3 bits time wait.
bit2 to bit0	L2 to L0: Data length selection bits	These bits are used to specify a data length for transmission/reception data. Selecting "000_B" sets the data length to 8 bits. Selecting "001_B" sets the data length to 5 bits. Selecting "010_B" sets the data length to 6 bits. Selecting "011_B" sets the data length to 7 bits. Selecting "100_B" sets the data length to 9 bits. Note: Settings other than above are prohibited.

28.12.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

■ Reception Data Register (RDR)

Figure 28.12-5 illustrates the bit structure of the serial reception register (RDR).

Figure 28.12-5 Bit Structure of Reception Data Register (RDR)

bit15.....	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	Initial value
	D8	D7	D6	D5	D4	D3	D2	D1	D0	00000000_B
	R	R	R	R	R	R	R	R	R	

R: Read only

The reception data register (RDR) is a 9-bit data buffer register for serial data reception.

- A serial data signal sent to a serial input pin (SIN pin) is converted through the shift register and then stored in the reception data register (RDR).
- "0" is placed in upper bits, as shown below, in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	0	X	X	X	X	X	X	X	X
7 bits	0	0	X	X	X	X	X	X	X
6 bits	0	0	0	X	X	X	X	X	X
5 bits	0	0	0	0	X	X	X	X	X

(X indicates the reception data bit)

- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR). A reception interrupt request will be generated if reception interrupts have been enabled (SSR: RIE = 1).
- Read the reception data register (RDR) when the reception data full flag bit (SSR:RDRF) is "1". The reception data full flag bit (SSR:RDRF) is cleared to "0" automatically, when the serial reception data register (RDR) is read.
- If a reception error occurs (SSR:ORE), the data in the reception data register (RDR) becomes invalid.
- 16-bit access is used to read RDR for a 9-bit transfer.

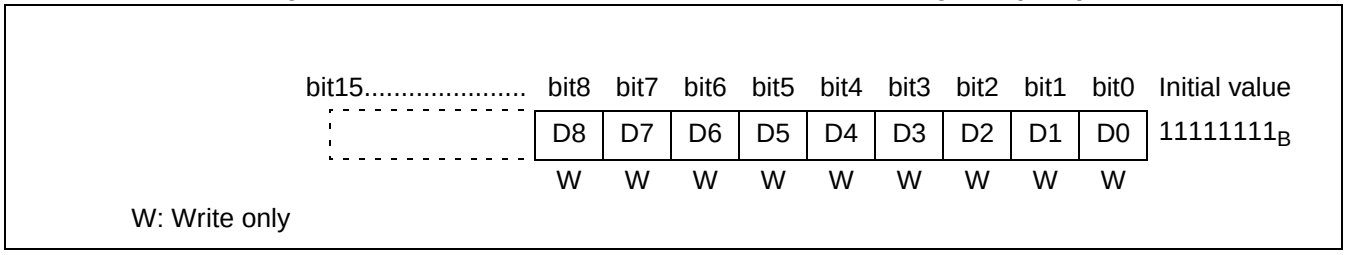
<Notes>

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.
- If a reception error occurs (SSR:ORE = 1) when the reception FIFO is used, the reception FIFO enable bit will be cleared and the reception data will not be stored to the reception FIFO.

■ Transmission Data Register (TDR)

Figure 28.12-6 illustrates the bit structure of the transmission data register.

Figure 28.12-6 Bit Structure of Transmission Data Register (TDR)



The transmission data register (TDR) is a 9-bit data buffer register for serial data transmission.

- If transmission data is written to the transmission data register (TDR) when transmission operation is enabled (SCR:TXE = 1), the transmission data will be transferred to the transmission shift register, converted into serial data and then sent from a serial data output pin (SOUT pin).
- As shown below, data becomes invalid from the upper bit in accordance with the data length.

Data length	D8	D7	D6	D5	D4	D3	D2	D1	D0
9 bits	X	X	X	X	X	X	X	X	X
8 bits	Invalid	X	X	X	X	X	X	X	X
7 bits	Invalid	Invalid	X	X	X	X	X	X	X
6 bits	Invalid	Invalid	Invalid	X	X	X	X	X	X
5 bits	Invalid	Invalid	Invalid	Invalid	X	X	X	X	X

(X indicates the transmission data bit)

- The transmission data empty flag (SSR:TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- If the transmission FIFO is disabled or empty, the transmission data empty flag (SSR:TDRE) will be set to "1" when transmission data is transferred to the transmission shift register and the transmission starts.
- The next transmission data can be written when the transmission data empty flag (SSR:TDRE) is set to "1". A transmission interrupt will occur if transmission interrupts have been enabled. Write the next transmission data by generating a transmission interrupt or when the transmission data empty flag (SSR:TDRE) is set to "1".

- Transmission data cannot be written to the transmission data register (TDR) when the transmission data empty flag (SSR:TDRE) is set to "0" and the transmission FIFO is either disabled or full.
- 16-bit access is used to write to TDR for a 9-bit transfer.

<Notes>

- The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.
- For information about the timing for setting the transmission data empty flag (SSR:TDRE) when the transmission FIFO is used, refer to Section "28.13.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing".

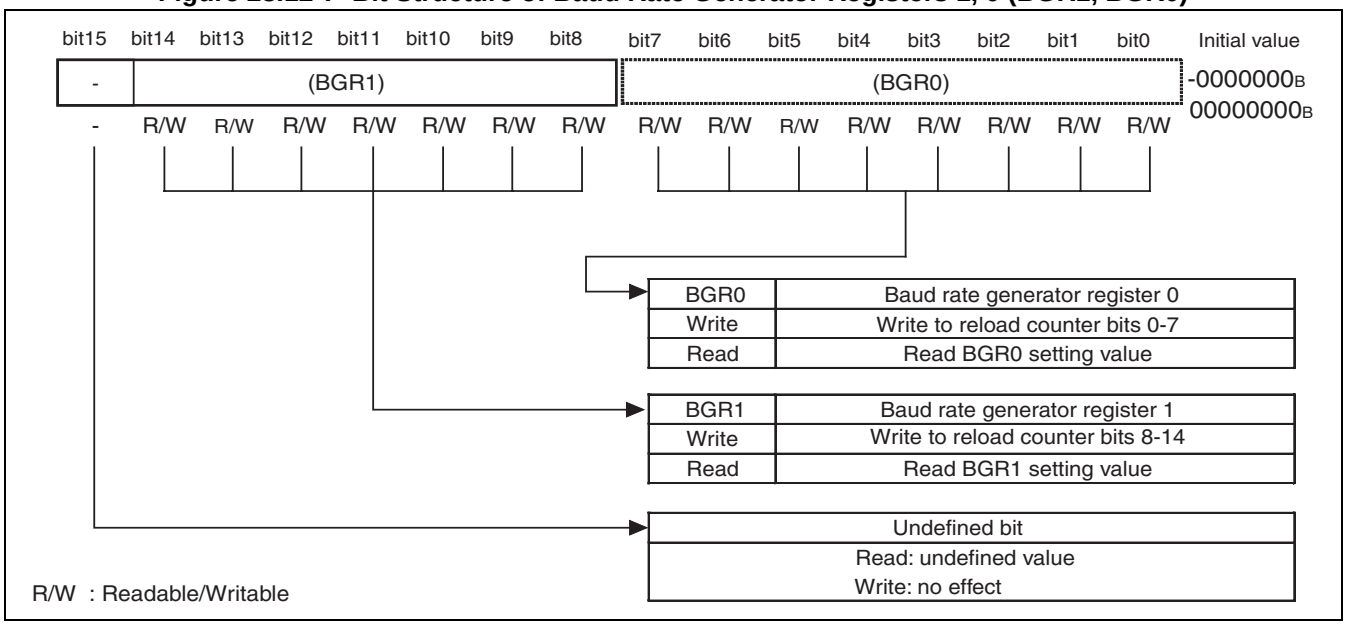
28.12.6 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 28.12-7 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 28.12-7 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
- (BGR1) (BGR0) -0000000B 00000000B
- R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
BGR0 Baud rate generator register 0
Write Write to reload counter bits 0-7
Read Read BGR0 setting value
BGR1 Baud rate generator register 1
Write Write to reload counter bits 8-14
Read Read BGR1 setting value
Undefined bit
Read: undefined value
Write: no effect
R/W : Readable/Writable

- A value is set to the baud rate generator registers 1, 0 (BGR1, BGR0).
- BGR0 and BGR1 correspond to the lower bits and upper bits respectively and they can write a reload value to be counted as well as read BGR0/BGR1 setting values.
- The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

<Notes>

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the setting of the SCINV bit. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
  - SCINV = 0: The "H" width of the serial clock is 1 peripheral clock (PCLK) cycle longer.
  - SCINV = 1: The "L" width of the serial clock is 1 peripheral clock (PCLK) cycle longer.
- Select 1 or a larger number for the reload value. However, select 3 or a larger value for the reload value of the CSIO which will become the master, when using these CSIO's as the master and slave.
- When a setting value of the baud rate generator registers 1, 0 (BGR1, BGR0) is changed, the new setting value is not reloaded until the counter value becomes "0000_H". To make the new setting value valid immediately, therefore, execute a CSIO reset (UPCL) after changing the BGR0/BGR1 setting value.
- Set a baud rate to BGR0/BGR1 during the use of reception FIFO to set the reception FIFO idle detection enable bit (FCR1:FRIIE) to "1" and operate in slave mode.

28.12.7 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 28.12-8 shows the bit structure of the FIFO control register 1 (FCR1) and Table 28.12-7 describes the function of each bit.

Figure 28.12-8 Bit Structure of FIFO Control Register 1 (FCR1)

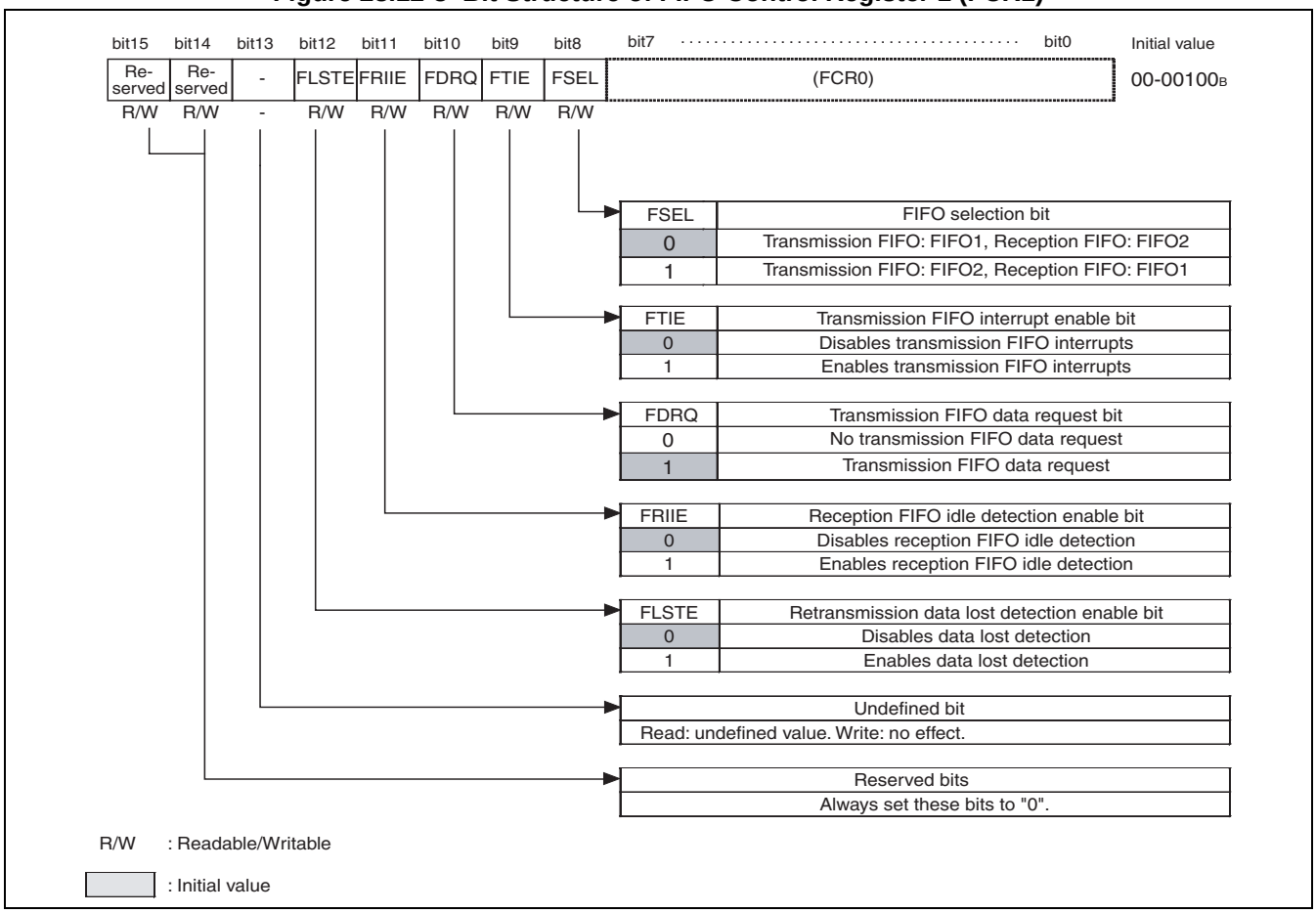

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
Re-served Re-served - FLSTE FRIIE FDRQ FTIE FSEL (FCR0) 00-00100B
R/W R/W - R/W R/W R/W R/W R/W
FSEL FIFO selection bit
0 Transmission FIFO: FIFO1, Reception FIFO: FIFO2
1 Transmission FIFO: FIFO2, Reception FIFO: FIFO1
FTIE Transmission FIFO interrupt enable bit
0 Disables transmission FIFO interrupts
1 Enables transmission FIFO interrupts
FDRQ Transmission FIFO data request bit
0 No transmission FIFO data request
1 Transmission FIFO data request
FRIIE Reception FIFO idle detection enable bit
0 Disables reception FIFO idle detection
1 Enables reception FIFO idle detection
FLSTE Retransmission data lost detection enable bit
0 Disables data lost detection
1 Enables data lost detection
Undefined bit
Read: undefined value. Write: no effect.
Reserved bits
Always set these bits to "0".
R/W : Readable/Writable
: Initial value

Table 28.12-7 Functional Description of Each Bit of FIFO Control Register 1 (FCR1)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "00_B".
bit13	Undefined bit	Read: undefined value Write: no effect
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission FIFO interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition • Writing "0" to this bit • When the transmission FIFO is full. Note: It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. Note: This bit cannot be cleared by resetting FIFO (FCR0:FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FCR0: FE2, FE1 = 0) in advance.

28.12.8 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 28.12-9 shows the bit structure of the FIFO control register 0 (FCR0) and Table 28.12-8 describes the function of each bit.

Figure 28.12-9 Bit Structure of FIFO Control Register 0 (FCR0)

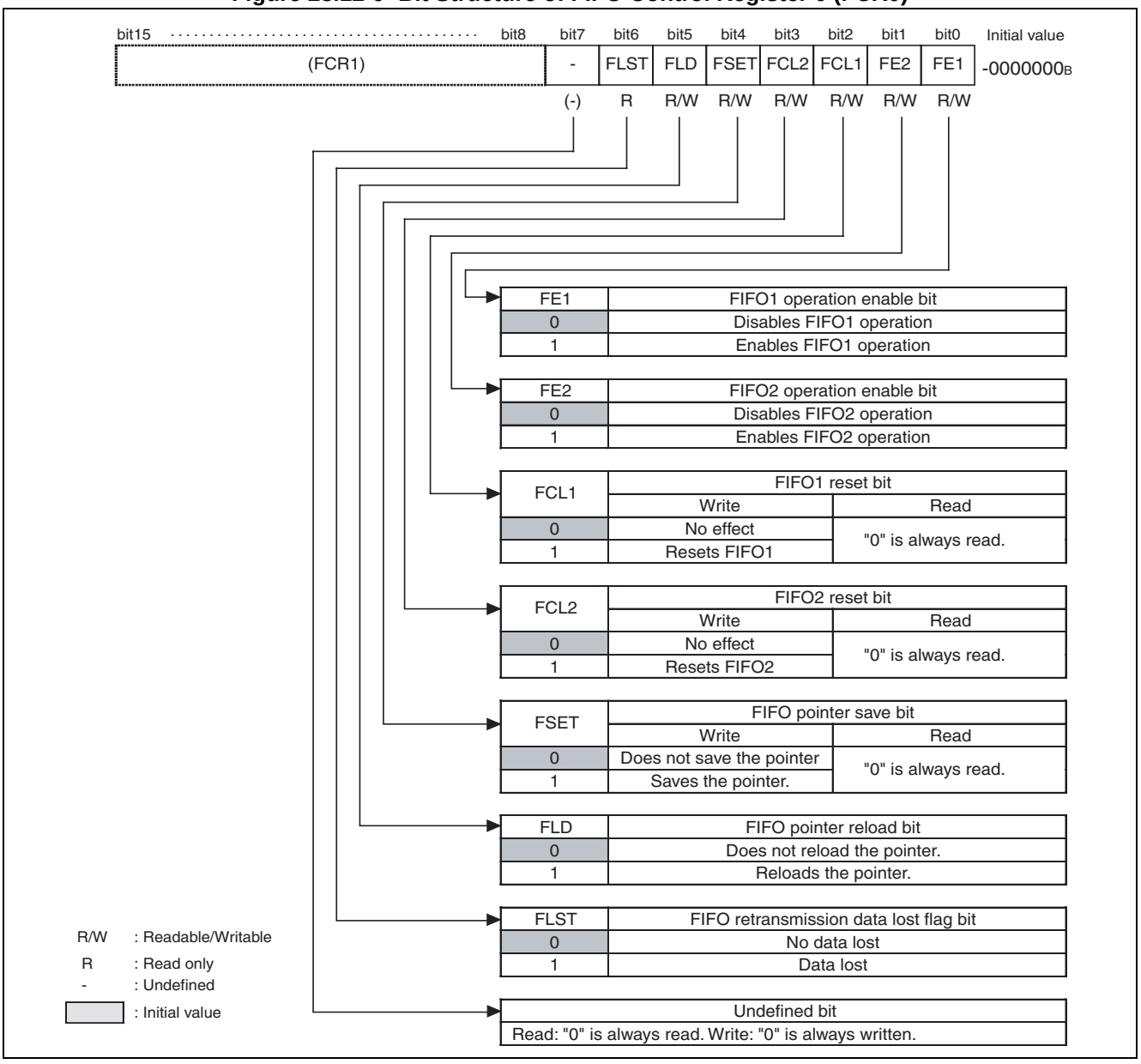

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FCR1) - FLST FLD FSET FCL2 FCL1 FE2 FE1 -0000000B
(-) R R/W R/W R/W R/W R/W R/W R/W
FE1 FIFO1 operation enable bit
0 Disables FIFO1 operation
1 Enables FIFO1 operation
FE2 FIFO2 operation enable bit
0 Disables FIFO2 operation
1 Enables FIFO2 operation
FCL1 FIFO1 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO1
FCL2 FIFO2 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO2
FSET FIFO pointer save bit
Write Read
0 Does not save the pointer "0" is always read.
1 Saves the pointer.
FLD FIFO pointer reload bit
0 Does not reload the pointer.
1 Reloads the pointer.
FLST FIFO retransmission data lost flag bit
0 No data lost
1 Data lost
Undefined bit
Read: "0" is always read. Write: "0" is always written.
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.12-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 2)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition • Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions • FIFO reset (writing "1" to FCL) • Writing "1" to the FLST bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. Note: Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE and TBIE bits to "0". And then, set the TIE and TBIE bits to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. Note: Set this bit to "1" when the number of transmission bytes (FBYTE1/ FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR0: FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO2. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".

Table 28.12-8 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 2)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR1:FLST1 bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable transmission/reception before resetting FIFO1. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. • To use FIFO2, set this bit to "1". • When FIFO2 is set for the transmission FIFO (FCR1:FSEL = 1) and "1" is written to this bit, transmission will start immediately, if FIFO2 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO2 is disabled, its status is retained.
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. • To use FIFO1, set this bit to "1". • When FIFO1 is set for the transmission FIFO (FCR1:FSEL = 0) and "1" is written to this bit, transmission will start immediately, if FIFO1 contains data and the UART is enabled for transmission (TXE = 1). At this point, set the TIE and TBIE bits to "0", write "1" to this bit, and then set the TIE and TBIE bits to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission buffer is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception buffer is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO1 is disabled, its status is retained.

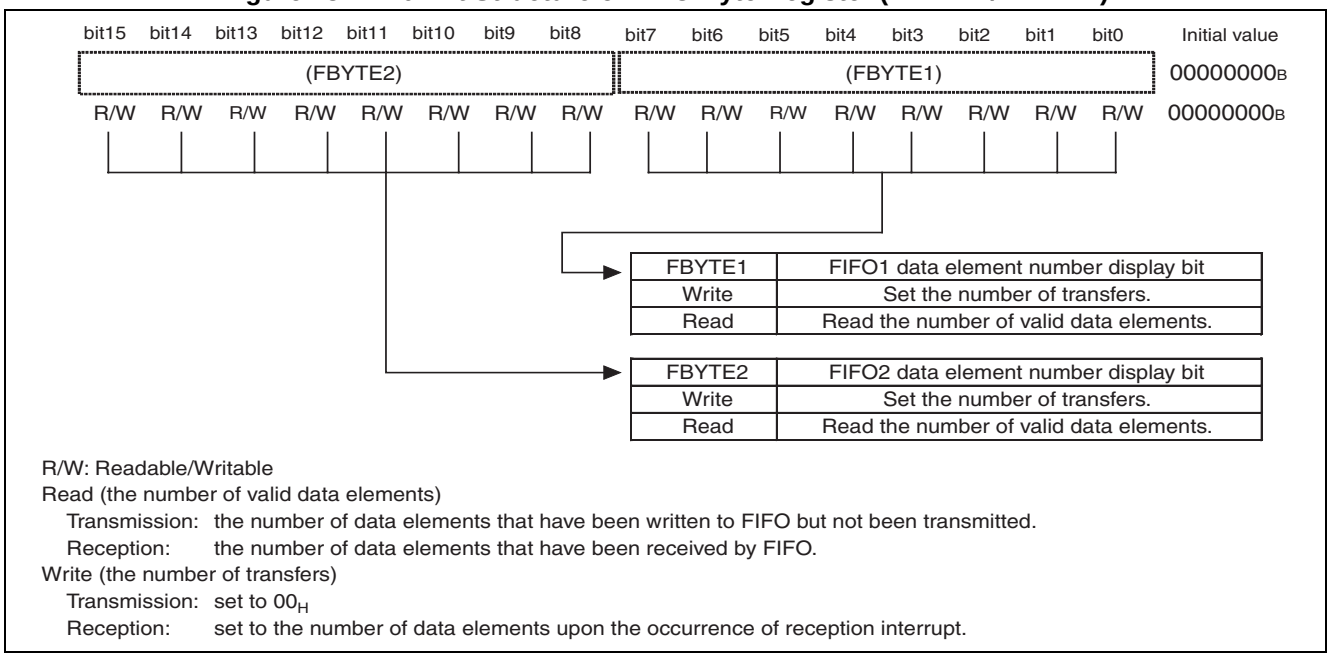
28.12.9 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 28.12-10 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 28.12-10 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FBYTE2)
(FBYTE1)
00000000B
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
00000000B
FBYTE1 FIFO1 data element number display bit
Write Set the number of transfers.
Read Read the number of valid data elements.
FBYTE2 FIFO2 data element number display bit
Write Set the number of transfers.
Read Read the number of valid data elements.
R/W: Readable/Writable
Read (the number of valid data elements)
Transmission: the number of data elements that have been written to FIFO but not been transmitted.
Reception: the number of data elements that have been received by FIFO.
Write (the number of transfers)
Transmission: set to 00H
Reception: set to the number of data elements upon the occurrence of reception interrupt.

The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the setting of the FCR1:FSEL bit.

Table 28.12-9 Displaying the Number of Data Elements

FSEL	FIFO selection	Number of bytes displayed
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is 08_H.
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.

- To receive data in master operation (master reception), set the TIE and TBIE bits to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. When the TXE bit is set to "1", the serial clock will be output for a specified amount of data so that the specified amount of data can be received. To set the TIE and TBIE bits to "1", wait until the FDRQ becomes "1".

<Notes>

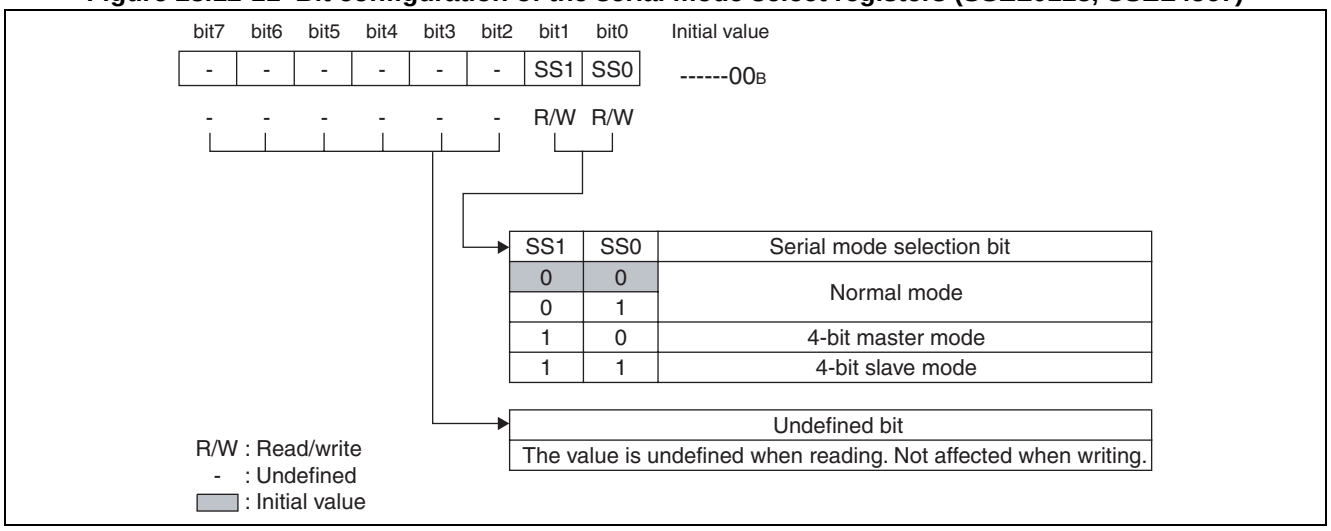
- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
- Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE and TBIE bits are set to "0".
- To disable reception (RXE = 0) while receiving data in master operation, disable transmission/ reception after disabling the transmission FIFO.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- To modify the FBYTE1/FBYTE2 of the reception FIFO, disable reception beforehand.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

28.12.10 Serial Mode Select Registers (SSEL0123, SSEL4567)

These registers operate the CSIO with 4-channels by using 1 clock simultaneously, enabling 4-bit serial communication.
The combination of ch.0 to ch.3 and that of ch.4 to ch.7 enable 4-channel simultaneous communication.

Figure 28.12-11 shows the bit configuration of the serial mode select registers (SSEL0123, SSEL4567).

Figure 28.12-11 Bit configuration of the serial mode select registers (SSEL0123, SSEL4567)


bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
- - - - - - SS1 SS0 ------00B
- - - - - - R/W R/W
SS1 SS0 Serial mode selection bit
0 0 Normal mode
0 1
1 0 4-bit master mode
1 1 4-bit slave mode
Undefined bit
The value is undefined when reading. Not affected when writing.
R/W : Read/write
- : Undefined
: Initial value

<Note>

Set these registers when the operation of the CSIO is stopped.

[bit7 to bit2]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit1, bit0]: SS1, SS0 (Serial mode select bit)

This bit determines whether to enable 4-channel simultaneous CSIO communication. In addition, it also selects an operation mode in the case of 4-channel simultaneous communication.

The operation modes are as follows:

- Normal mode: 4-channel simultaneous communication is not used.
- 4-bit master mode: 4-channel simultaneous communication using either ch.0 to ch.3 or ch.4 to ch.7 is allowed in master mode.
- 4-bit slave mode: 4-channel simultaneous communication using either ch.0 to ch.3 or ch.4 to ch.7 is allowed in slave mode.

SS1	SS0	Explanation
0	0	Normal mode is set.
0	1	
1	0	4-bit master mode is set.
1	1	4-bit slave mode is set.

<Notes>

- To set as 4-bit master mode, make the following settings using the MS bit of the serial control register (SCR0 to SCR7).
  - ch.0 to ch.2/ch.4 to ch.6: Slave mode
  - ch.3/ch.7: Master mode
- To set as 4-bit slave mode, set slave mode for all channels used for simultaneous communication using the MS bit of the serial control register (SCR0 to SCR7).

28.12.11 Received Data Mirror Registers/Transmitted Data Mirror Registers (RDRM/TDRM)

The received data mirror register (RDRM) is a mirror register of the received data register (RDR) lower 8 bits.
The transmitted data mirror register (TDRM) is a mirror register of the transmitted data register (TDR) lower 8 bits.
Access to these registers allows access to the received data register (RDR) lower 8 bits the transmitted data register (TDR) lower 8 bits.
Use these registers when 4-channel simultaneous communication is used.

■ Received data mirror registers (RDRM)

The received data mirror register 0 (RDRM0) corresponds to lower 8 bits of the received data register 0 (RDR0) while the received data mirror register 7 (RDRM7) corresponds to lower 8 bits of the received data register 7 (RDR7).

The received data mirror registers (RDRM0 to RDRM7) of ch.0 to ch.3 or ch.4 to ch.7 are arranged in line. So, word access allows the registers to be read at one time. Use this for DMA transfer and other purposes.

For details, see "■ Operation in 4-channel Simultaneous Communication Mode" in "28.14 Operation of CSIO (Clock Synchronous Serial Interface)".

<Note>

When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

■ Transmitted data mirror registers (TDRM)

The transmitted data mirror register 0 (TDRM0) corresponds to lower 8 bits of the transmitted data register 0 (TDR0) while the transmitted data mirror register 7 (TDRM7) corresponds to lower 8 bits of the transmitted data register 7 (TDR7).

The transmitted data mirror registers (TDRM0 to TDRM7) of ch.0 to ch.3 or ch.4 to ch.7 are arranged in line. So, word access allows the registers to be written to at one time. Use this for DMA transfer and other purposes.

For details, see "■ Operation in 4-channel Simultaneous Communication Mode" in "28.14 Operation of CSIO (Clock Synchronous Serial Interface)".

<Note>

When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

28.13 Interrupts of CSIO (Clock Synchronous Serial Interface)

CSIO (clock synchronous serial interface) has the transmission/reception interrupt functionality. The following sources can be used to generate interrupt requests.

- When reception data is set in the reception data register (RDR) or a reception error occurs
- When transmission data is transferred from the transmission data register (TDR) to the transmission shift register and then transmission starts
- Transmission bus idle state (no transmission operation)
- Transmission FIFO data request

■ Interrupts of CSIO

Table 28.13-1 lists the interrupt control bits and interrupt sources of the CSIO.

Table 28.13-1 Interrupt Control Bits and Interrupt Sources of CSIO (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Reception	RDRF	SSR	Reception of 1 byte	SCR:RIE	Reading reception data (RDR)
			Reception of the amount of data specified in FBYTE1/FBYTE2 setting value		Reading reception data (RDR) until reception FIFO becomes empty
			Detection of the idle state of reception for 8 clocks with the baud rate clock or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	ORE	SSR	Overrun error		Writing "1" to the reception error flag clear bit (SSR:REC)

Table 28.13-1 Interrupt Control Bits and Interrupt Sources of CSIO (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Transmis-sion	TDRE	SSR	Transmission register being empty	SCR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	TBI	SSR	No transmission operation	SCR:TBIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit (FCR1:FDRQ), or transmission FIFO being full

*: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

28.13.1 Occurrence of Reception Interrupts and Flag Set Timing

Reception interrupts are generated by the completion of reception (SSR:RDRF) and the occurrence of a reception error (SSR:ORE).

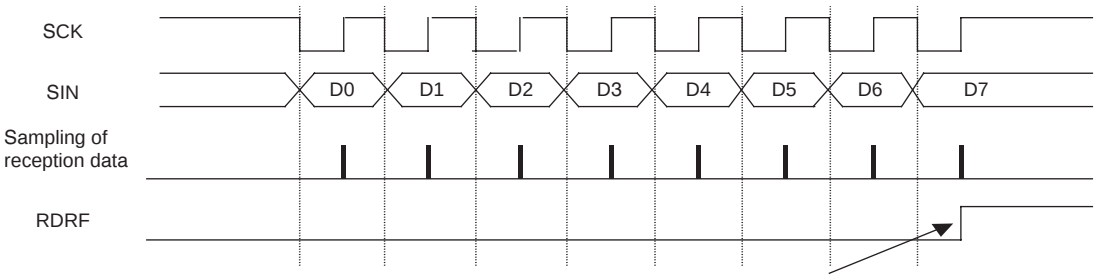
■ Occurrence of Reception Interrupts and Flag Set Timing

Reception data is stored to the reception data register (RDR) when the last data bit is detected. Each flag is set when the reception has been completed (SSR:RDRF = 1) or a reception error has occurred (SSR:ORE = 1). If reception interrupts have been enabled (SSR:RIE = 1), a reception interrupt will occur.

<Note>

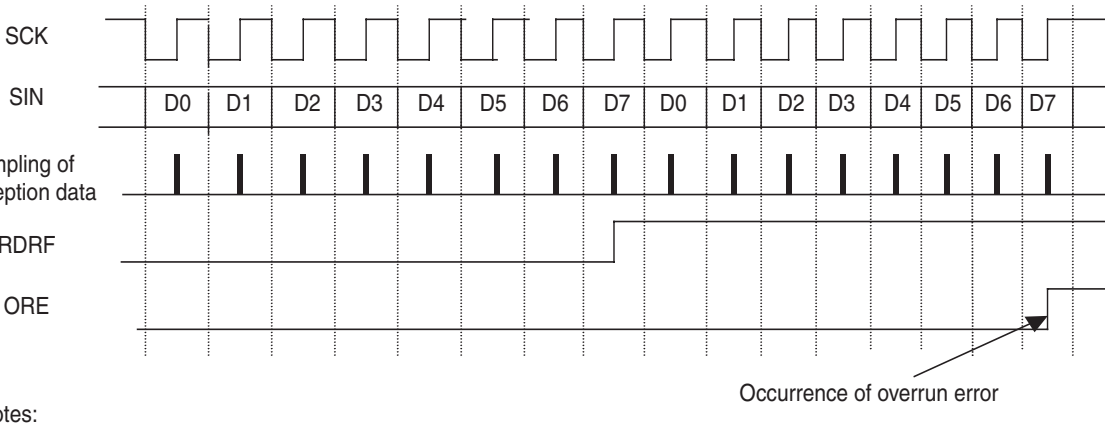
If a reception error occurs, the data in the reception data register (RDR) will become invalid.

Figure 28.13-1 Reception Operation and Flag Set Timing



Note:
The figure shows the timing based on the following conditions.
SCR: MS=1, SPI=0
ESCR: L2-L0=000_B
SMR: SCINV=0, BDS=0, SCKE=0, SOE=0

Figure 28.13-2 Timing for Setting ORE (Overrun Error) Flag



Notes:

The figure shows the timing based on the following conditions.
SCR: MS=1, SPI=0
ESCR: L2-L0=000_B
SMR: SCINV=0, BDS=0, SCKE=0, SOE=0

An overrun error occurs when the next data is transferred before reception data is read (RDRF = 1).

28.13.2 Occurrence of Interrupts when Reception FIFO is Used and Flag Set Timing

When the reception FIFO is used, an interrupt will occur, if the amount of data specified in the FBYTE1/FBYTE2 register (FBYTE1/FBYTE2) is received.

■ Occurrence of Reception Interrupts when Reception FIFO is Used and Flag Set Timing

Occurrence of interrupts when reception FIFO is used is determined by the setting value of the FBYTE1/FBYTE2 register.

- The reception data full flag of the serial status register (SSR:RDRF) is set to "1", when the received data is equivalent of the number of transfers specified in the FBYTE1/FBYTE2 register. At this point, a reception interrupt will occur, if reception interrupts have been enabled (SCR:RIE).
- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- The reception data full flag (SSR:RDRF) is cleared when reception data (RDR) is read until the reception FIFO becomes empty.
- An overrun error occurs (SSR:ORE = 1) when the next data is received while the number of valid reception data elements is indicating the capacity of FIFO.

Figure 28.13-3 Timing for Generating Reception Interrupt when Reception FIFO is Used

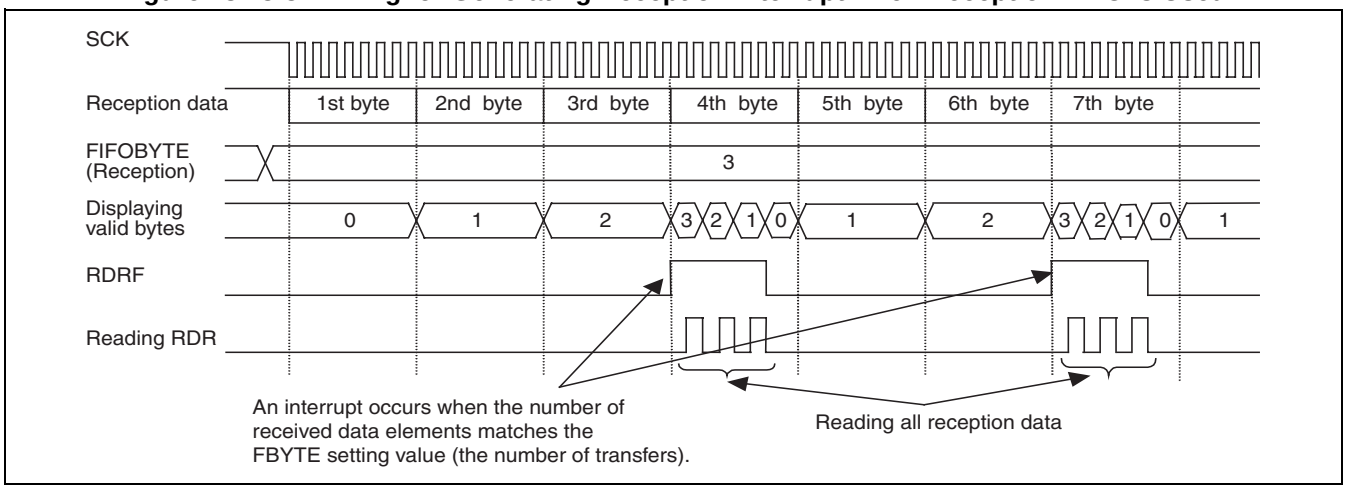
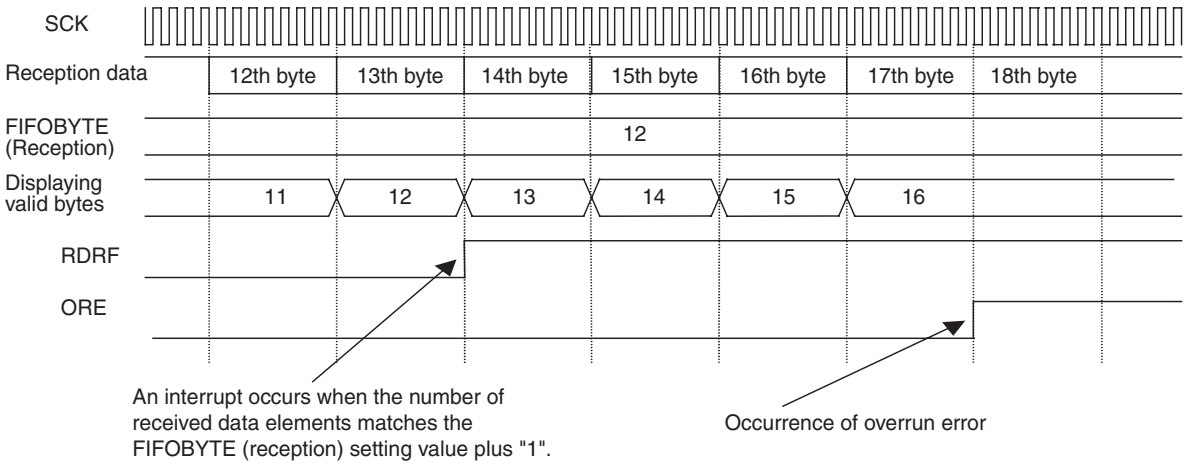


Figure 28.13-4 Timing for Setting ORE (Overrun Error) Flag Bit



Note: An overrun error occurs when the next data is received while the FIFO display is showing its capacity.

28.13.3 Occurrence of Transmission Interrupts and Flag Set Timing

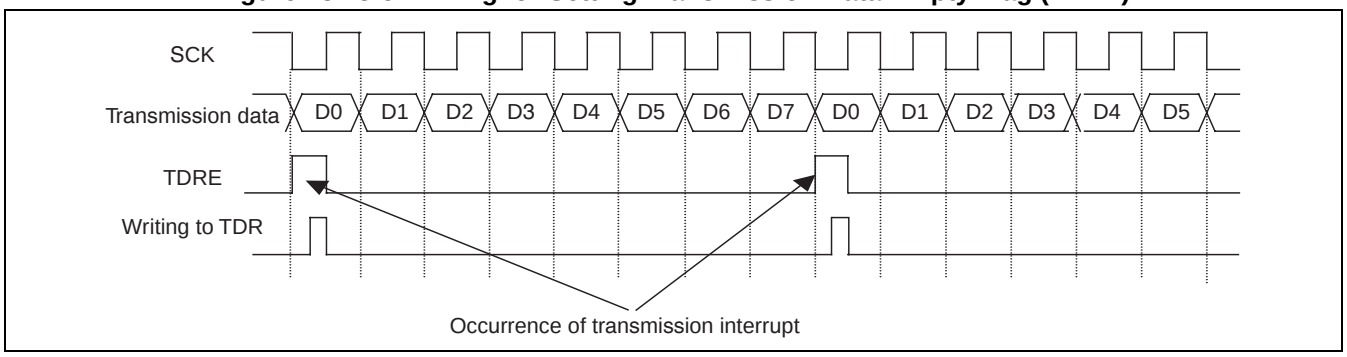
A transmission interrupt occurs when transmission data is transferred from the transmission data register (TDR) to the transmission shift register (SSR:TDRE = 1) and then the transmission starts, or when no transmission operation is in progress (SSR:TBI = 1).

■ Occurrence of Transmission Interrupts and Flag Set Timing

● Timing for setting the transmission data empty flag (TDRE)

It is enabled to write the next data (SSR:TDRE = 1), when the data written to the transmission data register (TDR) is transferred to the transmission shift register. At this point, a transmission interrupt will occur, if transmission interrupts have been enabled (SCR:TIE = 1). As the TDRE bit is a read only bit, it is cleared by writing "0" to the transmission data register (TDR).

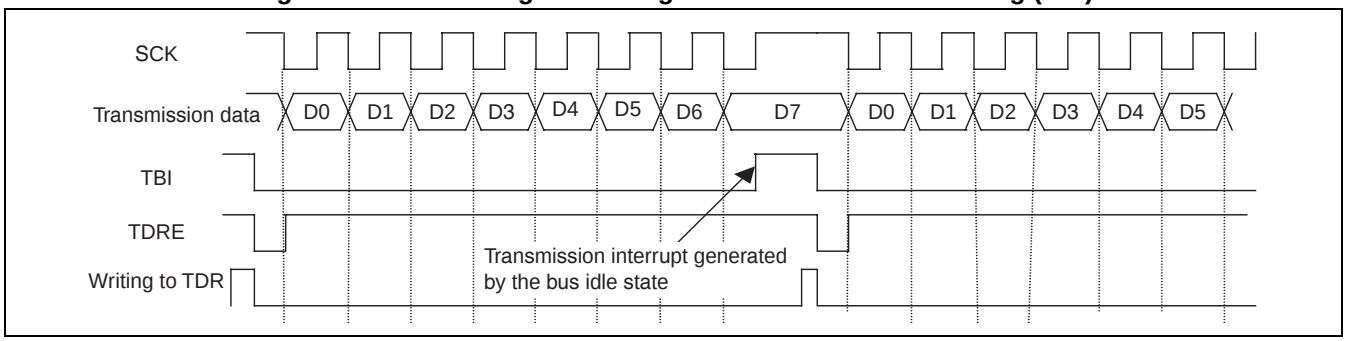
Figure 28.13-5 Timing for Setting Transmission Data Empty Flag (TDRE)



● Timing for setting the transmission bus idle flag (TBI)

The SSR:TBI bit is set to "1", when the transmission data register is empty (TDRE = 1) and no transmission operation is in progress. At this point, a transmission interrupt occurs if transmission bus idle interrupts have been enabled (SCR:TBIE = 1). The TBI bit and transmission interrupt request are cleared when transmission data is set to the transmission data register (TDR).

Figure 28.13-6 Timing for Setting Transmission Bus Idle Flag (TBI)



28.13.4 Occurrence of Interrupts when Transmission FIFO is Used and Flag Set Timing

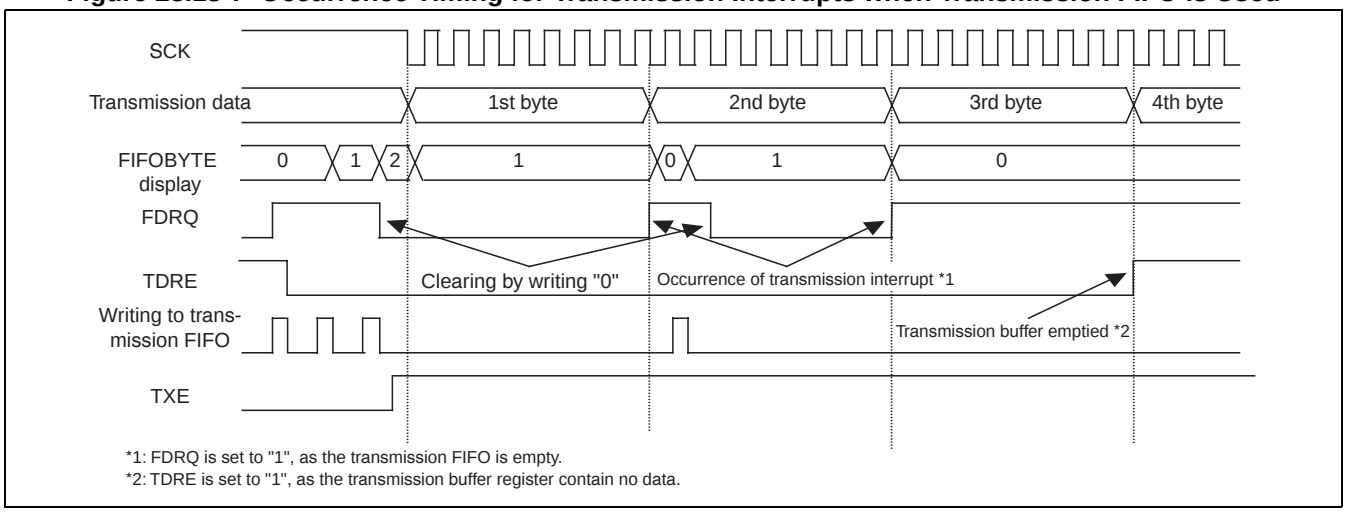
When the transmission FIFO is used, an interrupt will occur if the transmission FIFO does not contain any data.

■ Occurrence of Transmission Interrupts when Transmission FIFO is Used and Flag Set Timing

- The FIFO transmission data request bit (FCR1:FDRQ) is set to "1", when the transmission FIFO contains no data. At this point, a transmission interrupt will occur if FIFO transmission interrupts have been enabled (FCR1:FTIE = 1).
- Write "0" to the FIFO transmission data request bit (FCR1:FDRQ) to clear the interrupt request when required data has been written to the transmission FIFO upon the occurrence of a transmission interrupt.
- The FIFO transmission data request bit (FCR1:FDRQ) is set to "0" when the transmission FIFO becomes full.
- The FIFO byte register (FBYTE1/FBYTE2) can be read to check if the transmission FIFO contains any data.

  FBYTE1/FBYTE2 = 00_H indicates that the transmission FIFO contains no data.

Figure 28.13-7 Occurrence Timing for Transmission Interrupts when Transmission FIFO is Used



28.14 Operation of CSIO (Clock Synchronous Serial Interface)

CSIO uses clock synchronization for its transfer system.

■ Operation of CSIO (Clock Synchronous Serial Interface)

■ Normal Transfer (I)

● Features

Table 28.14-1 Features of Normal Transfer (I)

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Timing for transmission data output	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The setting values of registers required for the normal transfer (I) are shown below.

Table 28.14-2 Register Settings for Normal Transfer (I)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"
0: Set to "0"
*: User-defined setting

<Note>

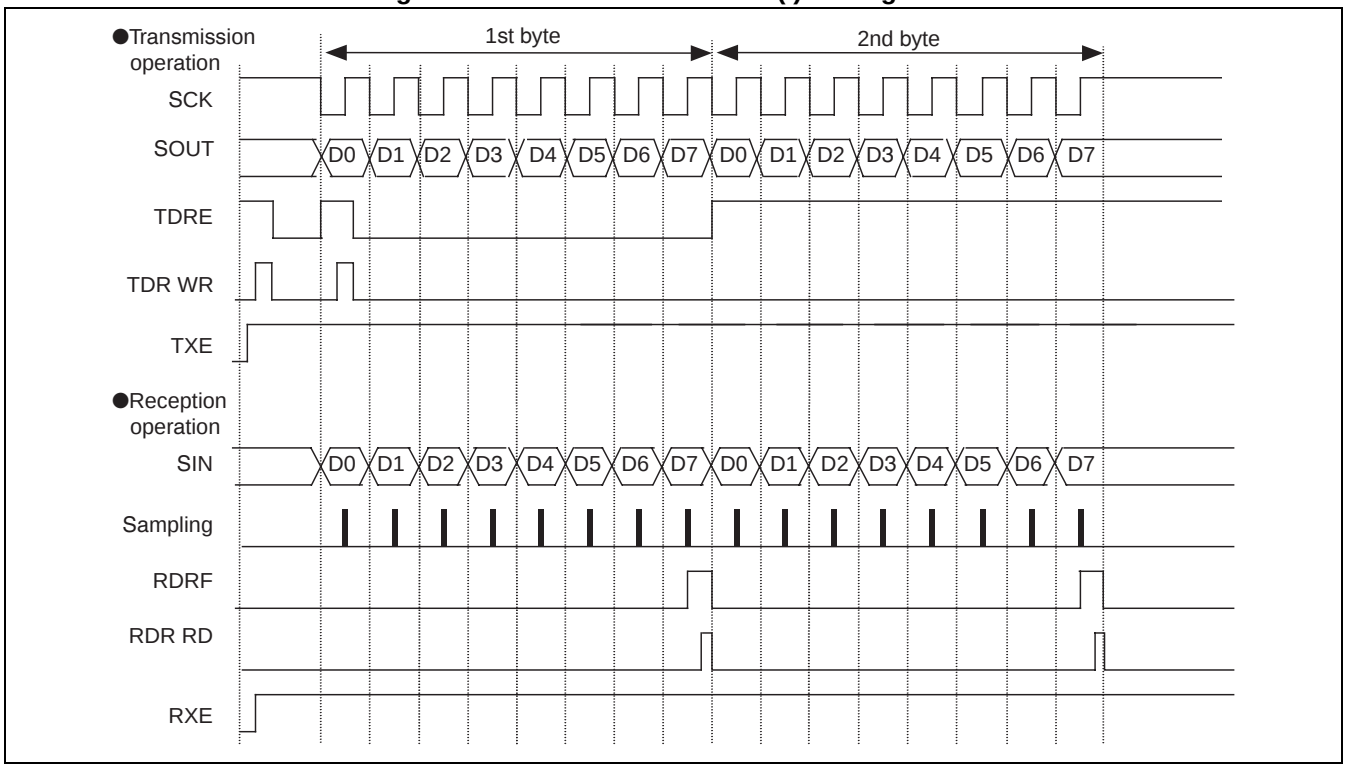
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● Normal transfer (I) timing chart

Figure 28.14-1 Normal Transfer (I) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

(1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.

(2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

<Notes>

- To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
- A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

- Transmission/reception operation

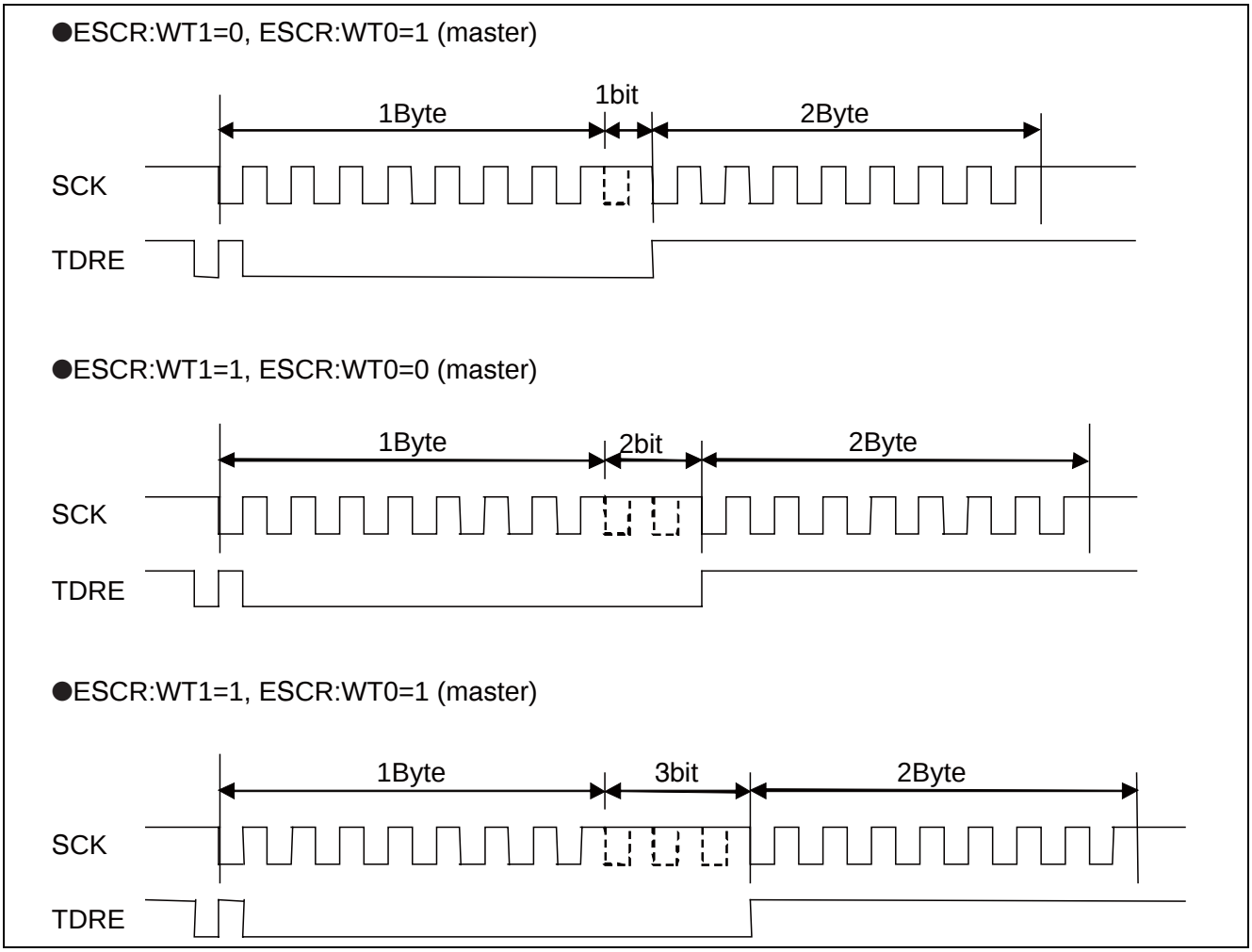
  (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

  (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

  (3) Reception data will be sampled at the rising edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Sequential data transmission/reception wait operation

  (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input.

  (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE=1), reception data will be sampled at the rising edge of the serial clock input (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation

(1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

(2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

(3) Reception data will be sampled at the rising edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

■ Normal Transfer (II)

● Features

Table 28.14-3 Features of Normal Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The table below shows the register setting values required for the normal transfer (II).

Table 28.14-4 Register Settings for Normal Transfer (II)

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	0	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

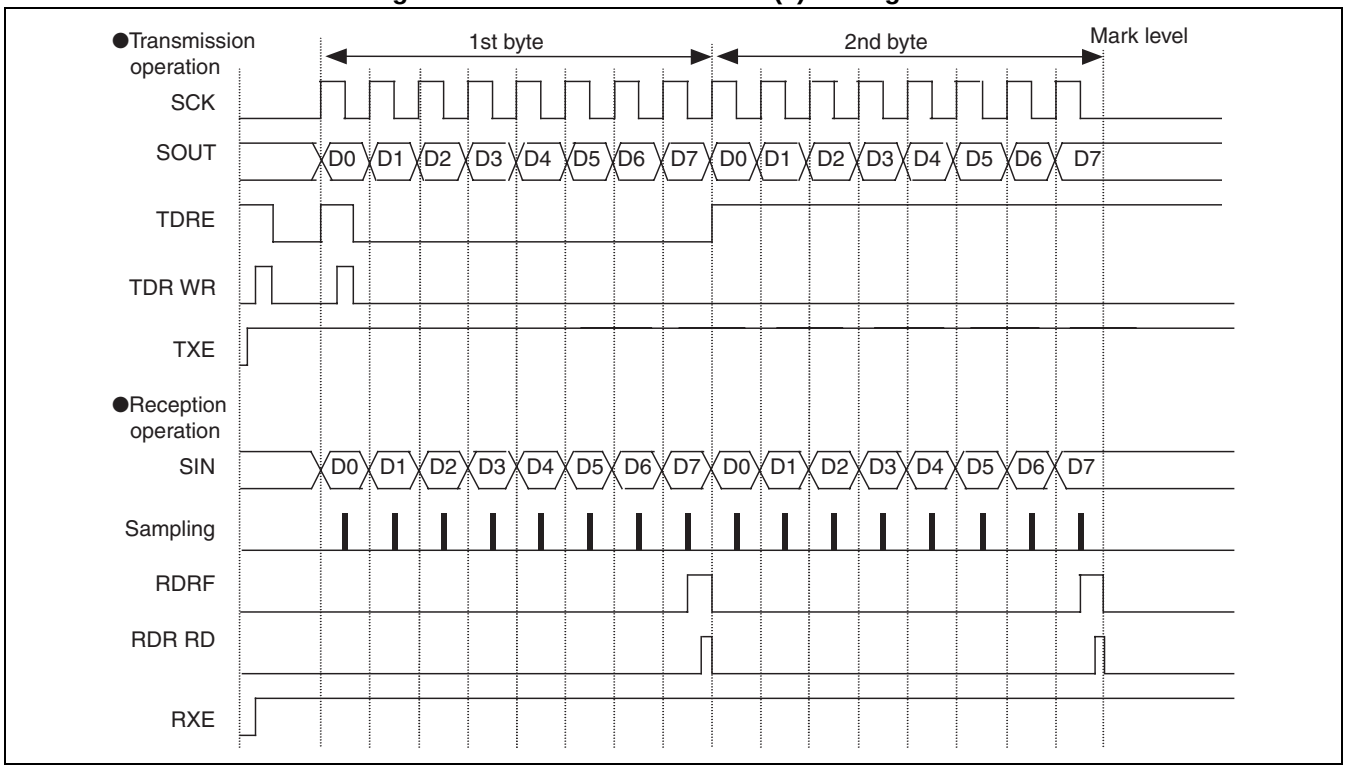
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

■ Normal Transfer (II) Timing Chart

Figure 28.14-2 Normal Transfer (II) Timing Chart



● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

(1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.

(2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

<Notes>

- To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
- A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

- Transmission/reception operation

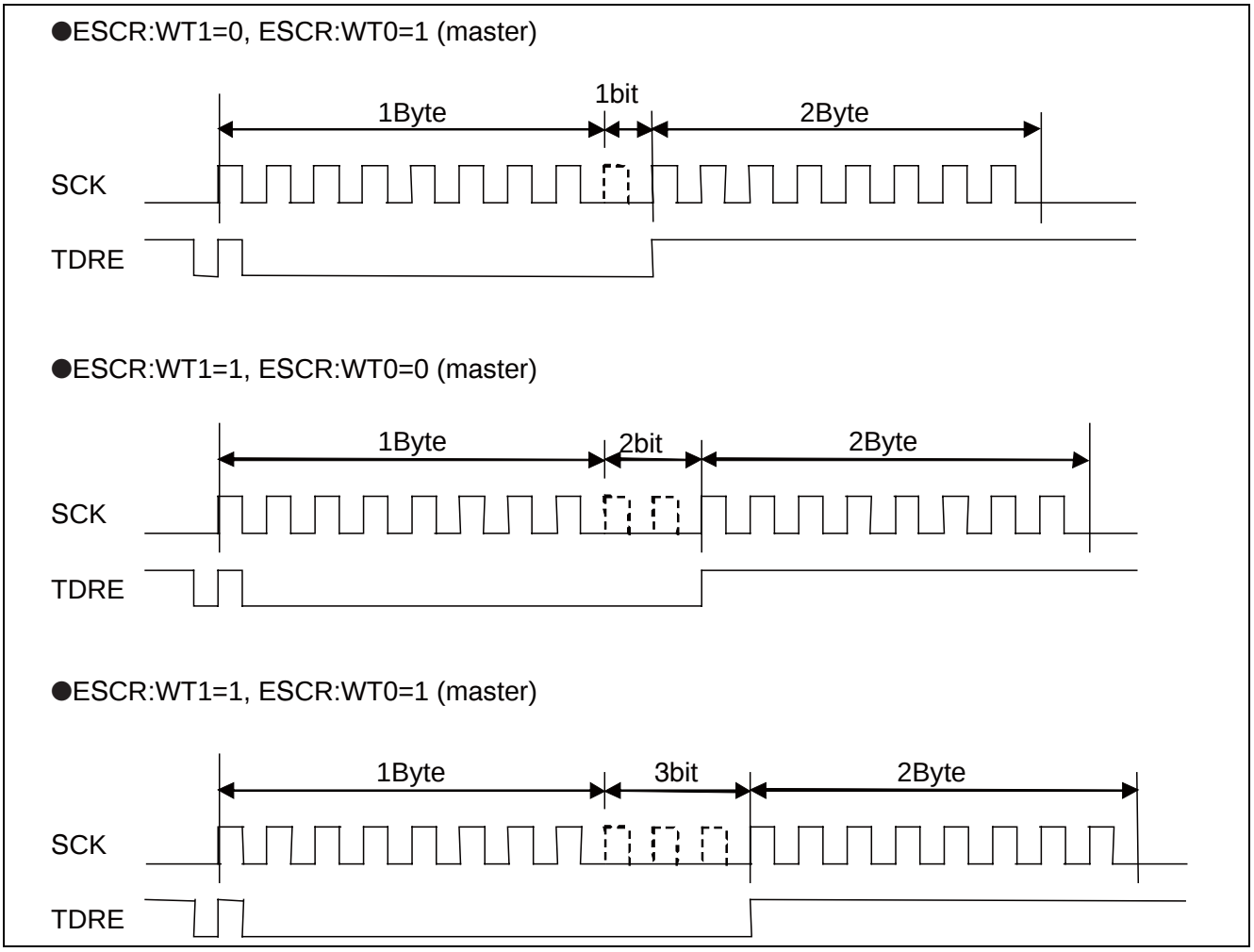
  (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

  (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

  (3) Reception data will be sampled at the falling edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Sequential data transmission/reception wait operation

  (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input.

  (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operations is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation

(1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

(2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

(3) Reception data will be sampled at the falling edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

■ SPI Transfer (I)

● Features

	Item	Description
1	Mark level of serial clock (SCK)	"H"
2	Output timing for transmission data	Rising edge of SCK
3	Sampling of reception data	Falling edge of SCK
4	Data length	5 bits to 9 bits

● Register settings

The table below shows the register setting values required for the SPI transfer (I).

Table 28.14-5 SPI Transfer (I) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	0	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

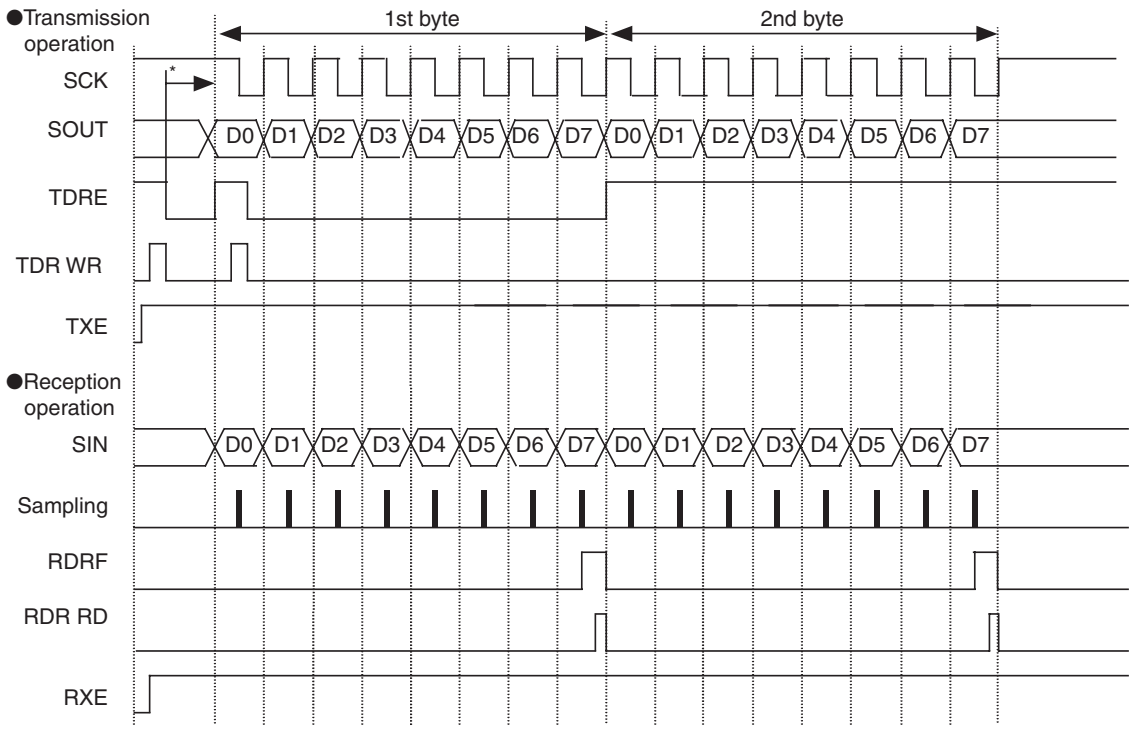
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation :SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● SPI transfer (I) timing chart

Figure 28.14-3 SPI Transfer (I) Timing Chart



* : During slave transmission (MS=1, SCKE=0, SCE=1), a duration of 4 or more peripheral clocks (PCLK) is required after data is written to TDR.

● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

(1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.

(2) SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock output (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1).
  At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

<Notes>

- To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
- A specified number of frames of the serial clock (SCK) will be output if the number of frames of the clock to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

- Transmission/reception operation

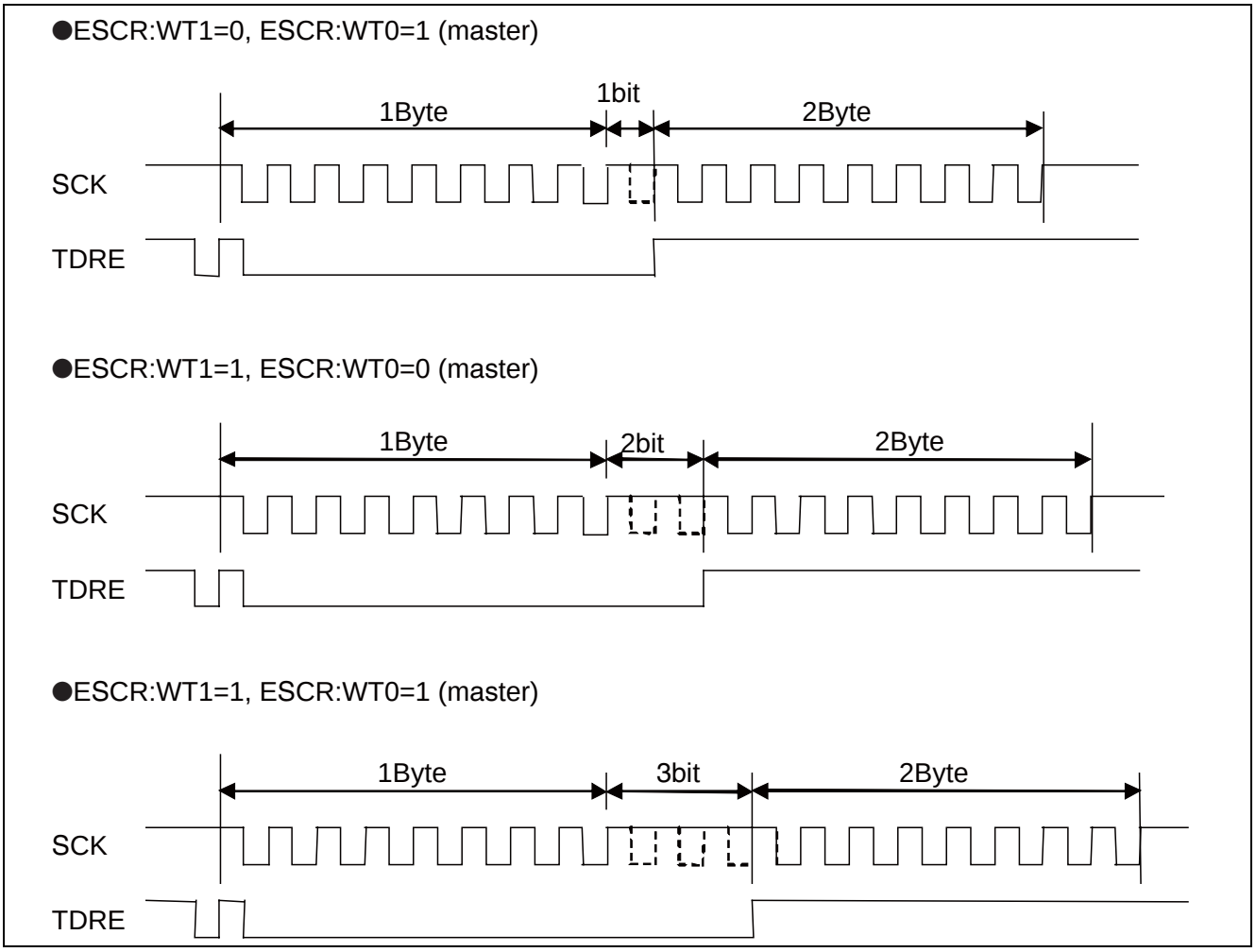
  (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

  (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output. SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

  (3) Reception data will be sampled at the falling edge of the serial clock (SCK) output. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Sequential data transmission/reception wait operation

  (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) output.

  (2) SSR:TDRE is set to "1", half a cycle before the falling edge of the first serial clock. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the falling edge of the serial clock input (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation

(1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

(2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the rising edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

(3) Reception data will be sampled at the falling edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

■ SPI Transfer (II)

● Features

Table 28.14-6 Features of SPI Transfer (II)

	Item	Description
1	Mark level of serial clock (SCK)	"L"
2	Output timing for transmission data	Falling edge of SCK
3	Sampling of reception data	Rising edge of SCK
4	Data length	5 to 9 bits

● Register settings

The table below shows the register setting values required for the SPI transfer (II).

Table 28.14-7 SPI Transfer (II) Register Settings

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SCR/ SMR	UPCL	MS	SPI	RIE	TIE	TBIE	RXE	TXE	MD2	MD1	MD0	-	SCINV	BDS	SCKE	SOE
	0	1/0	1	*	*	*	*	*	0	1	0	0	1	*	1/0	*
SSR/ ESCR	REC	-	-	-	ORE	RDRF	TDRE	TBI	SOP	-	-	WT1	WT0	L2	L1	L0
	0	-	-	-	-	-	-	-	0	-	-	*	*	*	*	*
RDR/ TDR	-							D8	D7	D6	D5	D4	D3	D2	D1	D0
	-							*	*	*	*	*	*	*	*	*
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
	-	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

1: Set to "1"

0: Set to "0"

*: User-defined setting

<Note>

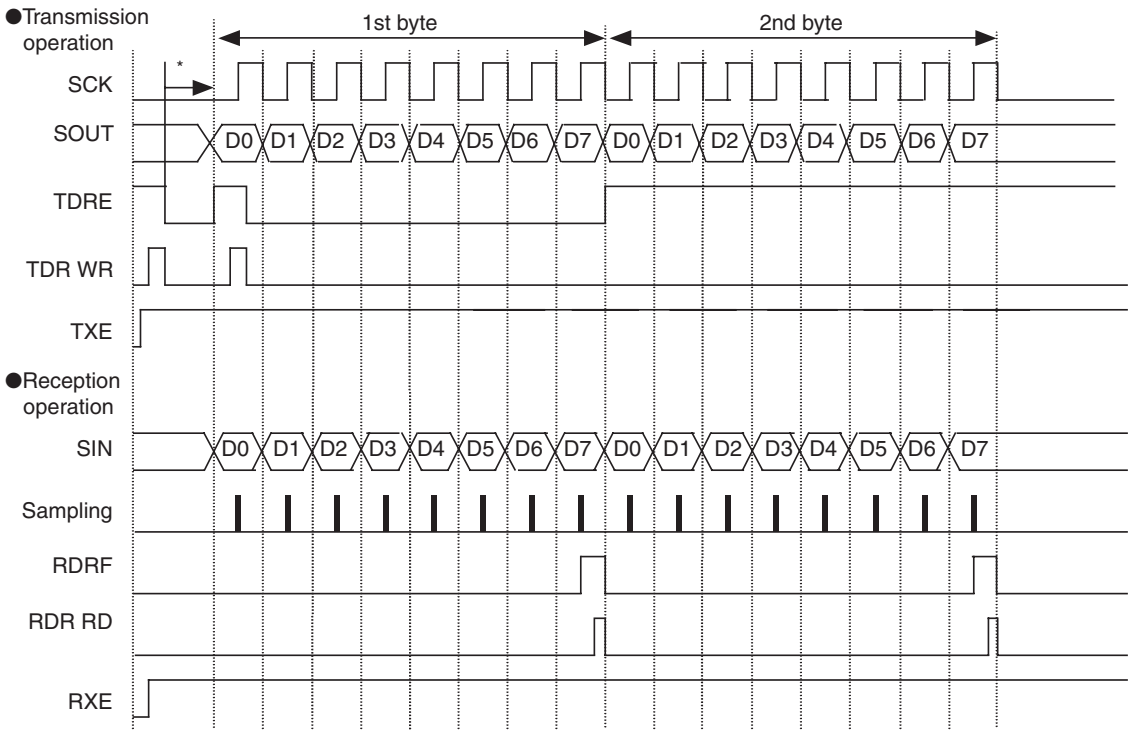
The setting values for the above bits (1/0) are different between master and slave operations. They must be set as shown below.

Master operation : SCR:MS = 0, SMR:SCKE = 1

Slave operation : SCR:MS = 1, SMR:SCKE = 0

● SPI transfer (II) timing chart

Figure 28.14-4 SPI Transfer (II) Timing Chart



*: During slave transmission (MS=1, SCKE=0, SCE=1), a duration of 4 or more peripheral clocks (PCLK) is required after data is written to TDR.

● Operational description

1. Master operation (SCR:MS = 0, SMR:SCKE = 1)

- Transmission operation

(1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1) but reception operation is disabled (SCR:RXE = 0), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output.

(2) SSR:TDRE is set to "1", half a cycle before the rising edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If dummy data is written to TDR when the output of serial data is disabled (SMR:SOE = 0) and both transmission and reception operations are enabled (SCR:TXE = 1, SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock output (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

<Notes>

- To only perform reception operation, write dummy data to TDR to output the serial clock (SCK).
- A specified number of frames of the serial clock (SCK) will be output if the byte number of frames to be transferred is set to the FBYTE1/FBYTE2 register when transmission/reception FIFO has been enabled.

- Transmission/reception operation

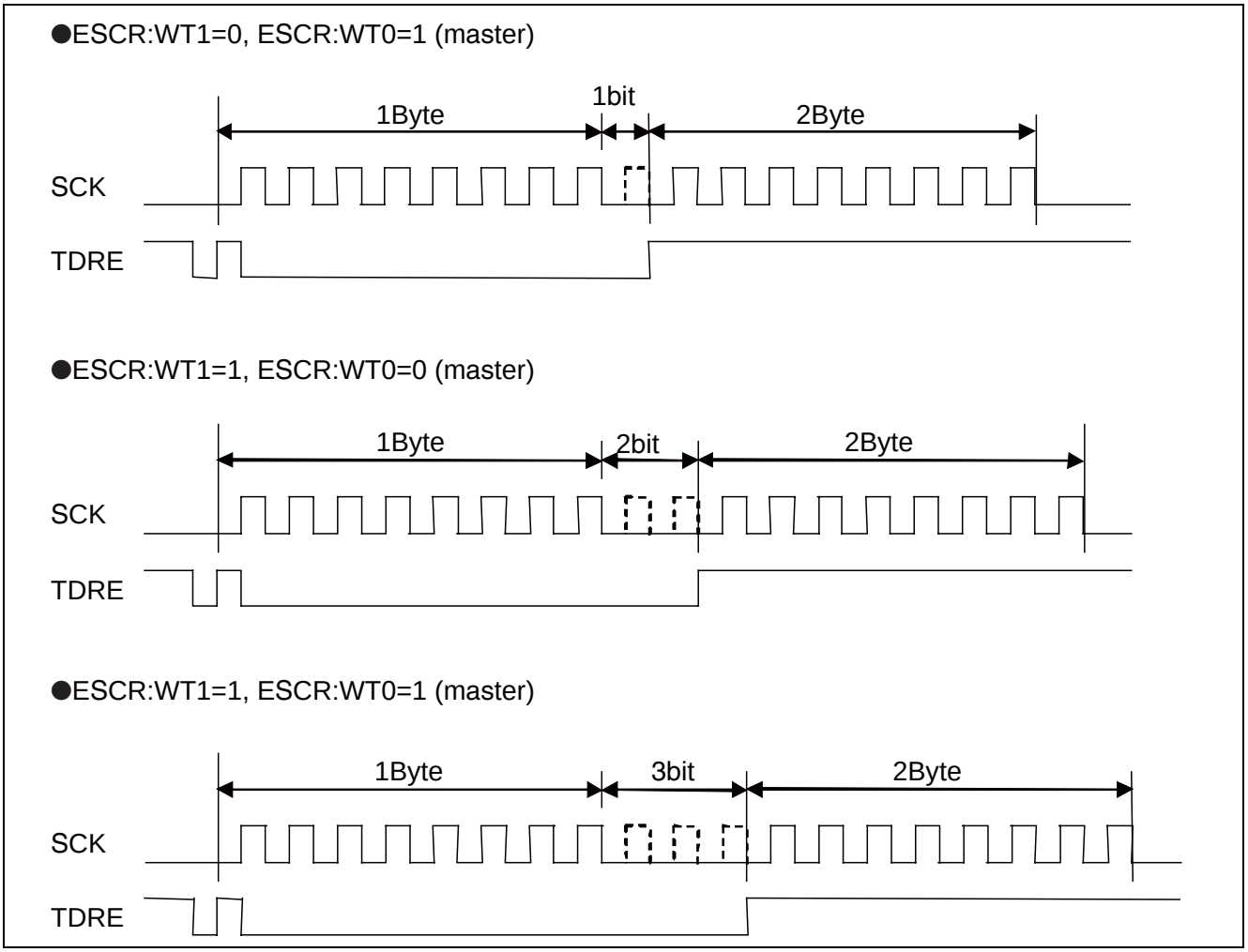
  (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

  (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) output. SSR:TDRE is set to "1", half a cycle before the rising edge of the first serial clock (SCK) output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

  (3) Reception data will be sampled at the rising edge of the transmission clock. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Sequential data transmission/reception wait operation

  (1) If signals other than (ESCR:WT1,ESCR:WT0) = (0,0) are set to sequential data transmission or reception, a wait is inserted between frames.



2. Slave operation (SCR:MS = 1, SMR:SCKE = 0)

- Transmission operation

  (1) If transmission data is written to TDR when the output of serial data is enabled (SMR:SOE = 1) and transmission operation is enabled (SCR:TXE = 1), SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input.

  (2) SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

- Reception operation

  (1) If the output of serial data is disabled (SMR:SOE = 0) and reception operation is enabled (SCR:RXE = 1), reception data will be sampled at the rising edge of the serial clock input (SCK).

  (2) SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read.

  (3) SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

- Transmission/reception operation

  (1) To perform transmission and reception operations at the same time, enable the output of serial data (SMR:SOE=1), and enable transmission/reception operation (SCR: TXE, RXE=1).

  (2) If transmission data is written to TDR, SSR:TDRE will be set to "0" and the first bit will be output. After that, the transmission data will be output, being synchronized with the falling edge of the serial clock (SCK) input. SSR:TDRE is set to "1" when the transmission data for the first bit is output. A transmission interrupt request is output when transmission interrupts have been enabled (SCR:TIE = 1). At this point, the transmission data for the second byte can be written.

  (3) Reception data will be sampled at the rising edge of the serial clock (SCK) input. SSR:RDRF is set to "1" when the last bit is received. A reception interrupt request is output when reception interrupts have been enabled (SCR:RIE = 1). At this point, reception data (RDR) can be read. SSR:RDRF is cleared to "0" once the reception data (RDR) is read.

■ Operation in 4-channel Simultaneous Communication Mode

It is possible to make the CSIO of 4-channels of either ch.0 to ch.3 or ch.4 to ch.7 communicate simultaneously to transmit and receive 4-bit data at one time.

The 4-channels can be used both in master mode and in slave mode. This section explains the operation in 4-channel simultaneous communication mode.

● Overview

To allow 4-channel simultaneous communication, setting is made by the SS1 and SS0 bit of the serial mode select register (SSEL0123, SSEL4567).

In addition, the required settings vary according to whether communication is executed in master mode or in slave mode.

Table 28.14-8 shows the settings required for the 4-channel simultaneous communication mode.

Table 28.14-8 Settings in 4-channel simultaneous communication mode

Mode	Setting		ch.0/ch.4	ch.1/ch.5	ch.2/ch.6	ch.3/ch.7
4-bit master	SSEL	SS1/SS0 bit	10	10	10	10
	SCR	MS bit	1	1	1	0
4-bit slave	SSEL	SS1/SS0 bit	11	11	11	11
	SCR	MS bit	1	1	1	1

SSEL: Serial mode select register (SSEL0123, SSEL4567)

SCR: Serial control register (SCR0 to SCR11)

The serial clock input methods vary according to whether the mode is 4-bit master mode or 4-bit slave mode.

Table 28.14-9 lists the input sources of the serial clock.

Table 28.14-9 Input Sources of the Serial Clock

Mode	ch.0 /ch.4	ch.1/ch.5	ch.2/ch.6	ch.3/ch.7
4-bit master (SS1, SS0 = 10)	Output from ch.3/ch.7	Output from ch.3/ch.7	Output from ch.3/ch.7	SCK3/SCK7 pin
4-bit slave (SS1, SS0 = 11)	SCK3/SCK7 pin	SCK3/SCK7 pin	SCK3/SCK7 pin	SCK3/SCK7 pin

Figure 28.14-5 shows the input sources of the serial clock in 4-bit master mode and in 4-bit slave mode.

Figure 28.14-5 The Input Sources of the Serial Clock

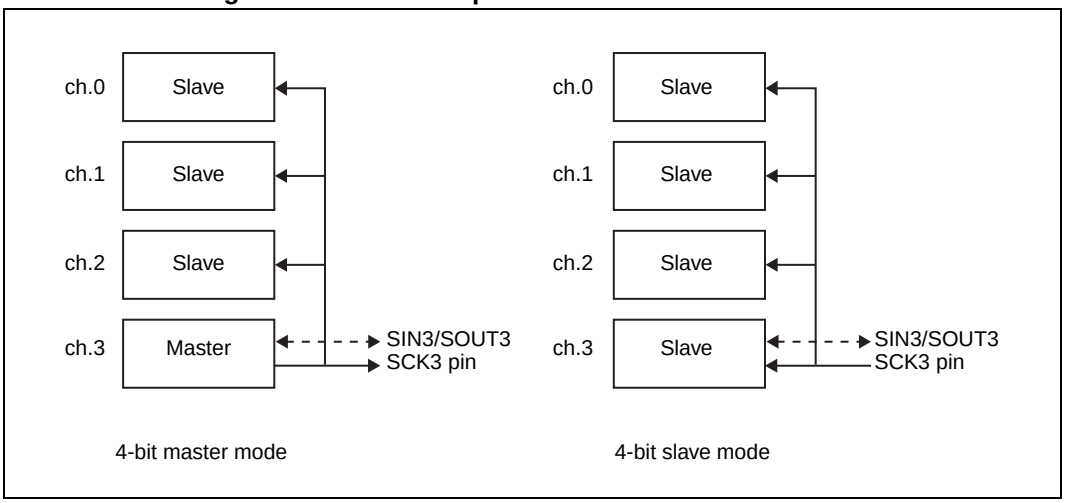


Table 28.14-10 shows the available pin combination for the four-channel simultaneous communication mode.

Table 28.14-10 Available Pin Combination

	ch.0 / ch.4	ch.1 / ch.5	ch.2 / ch.6	ch.3 / ch.7
Combination 1	SCK0_1 SIN0_1 SOUT0_1	SCK1 SIN1 SOUT1	SCK2 SIN2 SOUT2	SCK3 SIN3 SOUT3
Combination 2	SCK4 SIN4 SOUT4	SCK5 SIN5 SOUT5	SCK6 SIN6 SOUT6	SCK7 SIN7 SOUT7

● Operation

When 4-channel simultaneous communication mode is used, the receive operation/transmit operation is the same as the operation of 1 channel.

However, to allow 4-bit simultaneous transmission and reception, the following registers are provided.

- Received data mirror register (RDRM0 to RDRM7)
- Transmitted data mirror registers (TDRM0 to TDRM7)

Access to these registers allows access to the received data register (RDR) lower 8 bits or the transmitted data register (TDR) lower 8 bits.

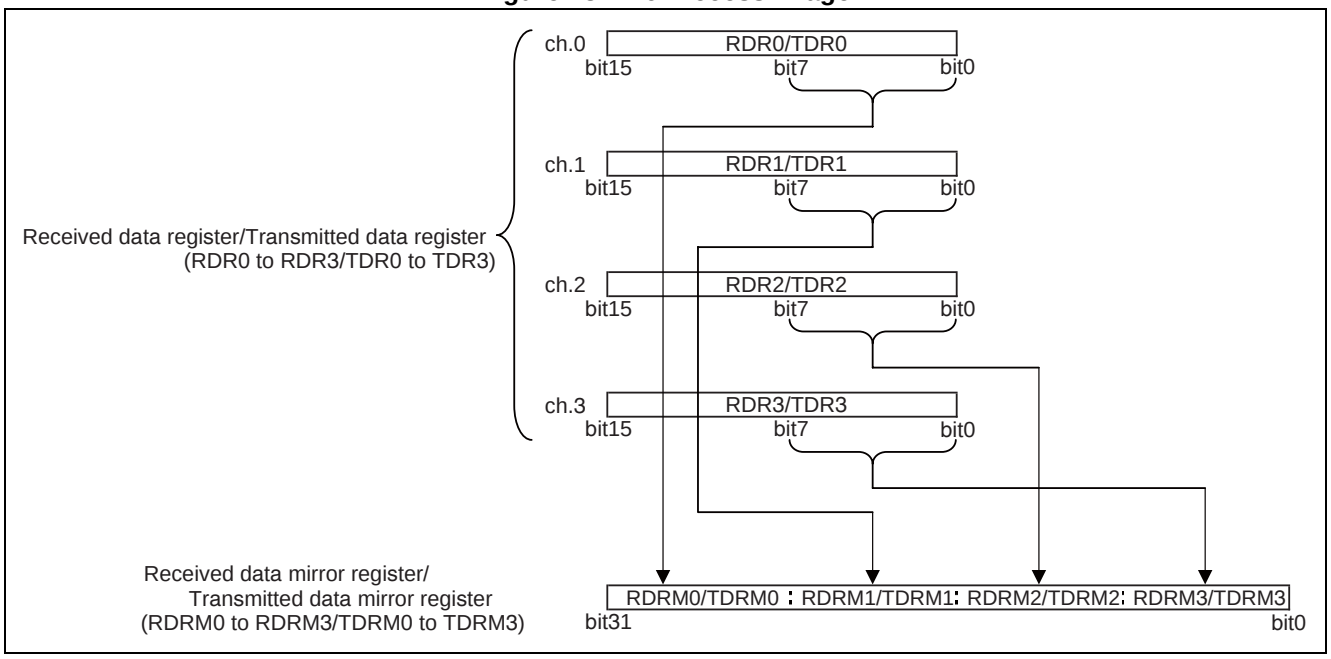
The received data mirror registers (RDRM0 to RDRM7)/transmitted data mirror registers (RDRM/TDRM) of ch.0 to ch.3 or ch.4 to ch.7 are arranged in line. So, word access allows the registers to be written to at one time. Use this for DMA transfer and other purposes.

<Note>

The interrupt during 4 channels are simultaneously activating, it is recommended to allow using 1 channel only out of 4 channels.

Figure 28.14-6 shows the images of the received data mirror register (RDRM0 to RDRM3)/transmitted data mirror register (TDRM0 to TDRM3).

Figure 28.14-6 Access image



<Note>

When 4-channel simultaneous communication is used, data with 9-bit length cannot be used.

28.15 Dedicated Baud Rate Generator

The dedicated baud rate generator only functions in master operation. However, set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

■ Baud Rate Selection for CSIO (Clock Synchronous Serial Interface)

The dedicated baud rage generator settings are different between master and slave operations.

● Master operation

The baud rate is selected by dividing the internal clock using the dedicated baud rate generator.

- There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).
- The reload counter divides the internal clock, according to the set value.

● Slave operation

In slave operation (SCR:MS = 1), the dedicated baud rate generator does not function.
(The slave operation directly uses the external clock which is input from the clock input pin SCK.)

<Note>

Set the dedicated baud rate generator even for slave operation, when using the reception FIFO.

28.15.1 Setting Baud Rate

This section describes how the baud rates are set and the resulting serial clock frequency is calculated.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$V = \phi / b - 1$

V: Reload value
b: Baud rate
ϕ: Peripheral clock (PCLK) frequency

(2) Example of calculation

If the peripheral clock (PCLK) is 16MHz, the internal clock is used, and the baud rate is 19200bps, the reload value will be:

Reload value:

$V = (16 \times 1000000)/19200 - 1 = 832$

So baud rate is:

$b = (16 \times 1000000)/(832+1) = 19208$ bps

(3) Baud rate error

The following formula is used to calculate a baud rate error.

Error (%) = (calculated value - target value) / target value $\times$ 100

Example: peripheral clock (PCLK) = 20MHz, target baud rate = 153600bps

Reload value = $(20 \times 1000000)/153600 - 1 = 129$

Baud rate (calculated value) = $(20 \times 1000000)/(129+1) = 153846$ (bps)

Error (%) = $(153846 - 153600)/153600 \times 100 = 0.16$ (%)

<Notes>

- The reload counter halts when the reload value is set to "0".
- When the reload value is even-numbered, the "H" and "L" widths of the serial clock are as shown below, depending on the SCINV bit settings. When the reload value is odd-numbered, the "L" width is the same as the "H" width.
  - When SCINV is set to "0", the "H" width of the serial clock is one peripheral clock (PCLK) cycle longer.
  - When SCINV is set to "1", the "L" width of the serial clock is one peripheral clock (PCLK) cycle longer.
- Select 3 or a larger value for the reload value.

■ Reload Values and Baud Rates for Different Peripheral Clock (PCLK) Frequencies

Table 28.15-1 Reload Values and Baud Rates

Baud rate (bps)	8 MHz		10 MHz		16 MHz		20 MHz		24 MHz		32MHz	
	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR	Value	ERR
8M	-	-	-	-	-	-	-	-	-	-	3	0
6M	-	-	-	-	-	-	-	-	3	0	-	-
5M	-	-	-	-	-	-	3	0	-	-	-	-
4M	-	-	-	-	3	0	4	0	5	0	7	0
2.5M	-	-	3	0	-	-	-	-	-	-	-	-
2M	3	0	4	0	7	0	9	0	11	0	15	0
1M	7	0	9	0	15	0	19	0	23	0	31	0
500000	15	0	19	0	31	0	39	0	47	0	63	0
460800	-	-	-	-	-	-	-	-	51	-0.16	-	-
250000	31	0	39	0	63	0	79	0	95	0	127	0
230400	-	-	-	-	-	-	-	-	103	-0.16	-	-
153600	51	-0.16	64	-0.16	103	-0.16	129	-0.16	155	-0.16	207	-0.16
125000	63	0	79	0	127	0	159	0	191	0	255	0
115200	68	-0.64	86	0.22	138	0.08	173	0.22	207	-0.16	277	0.08
76800	103	-0.16	129	-0.16	207	-0.16	259	-0.16	311	-0.16	416	0.08
57600	138	0.08	173	0.22	277	0.08	346	-0.16	416	0.08	555	0.08
38400	207	-0.16	259	-0.16	416	0.08	520	0.03	624	0	832	-0.04
28800	277	0.08	346	<0.01	554	-0.01	693	-0.06	832	-0.03	1110	-0.01
19200	416	0.08	520	0.03	832	-0.03	1041	0.03	1249	0	1666	0.02
10417	767	<0.01	959	<0.01	1535	<0.01	1919	<0.01	2303	<0.01	3071	<0.01
9600	832	0.04	1041	0.03	1666	0.02	2083	0.03	2499	0	3332	-0.01
7200	1110	<0.01	1388	<0.01	2221	<0.01	2777	<0.01	3332	<0.01	4443	-0.01
4800	1666	0.02	2082	-0.02	3332	<0.01	4166	<0.01	4999	0	6666	<0.01
2400	3332	<0.01	4166	<0.01	6666	<0.01	8332	<0.01	9999	0	13332	<-0.01
1200	6666	<0.01	8334	0.02	13332	<0.01	16666	<0.01	19999	0	26666	<0.01
600	13332	<0.01	16666	<0.01	26666	<0.01	-	-	-	-	-	-
300	26666	<0.01	-	-	-	-	-	-	-	-	-	-

- Value: the value set in BGR1/BGR0 registers
- ERR: baud rate error (%)

■ Functions of Reload Counters

There are two reload counters, a transmission reload counter and a reception reload counter, which function as a dedicated baud rate generator. Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock.

■ Starting a Count

The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

■ Restart

The reload counter restarts under the following conditions.

● For both transmission and reception reload counters

Programmable reset (SCR:UPCL bit)

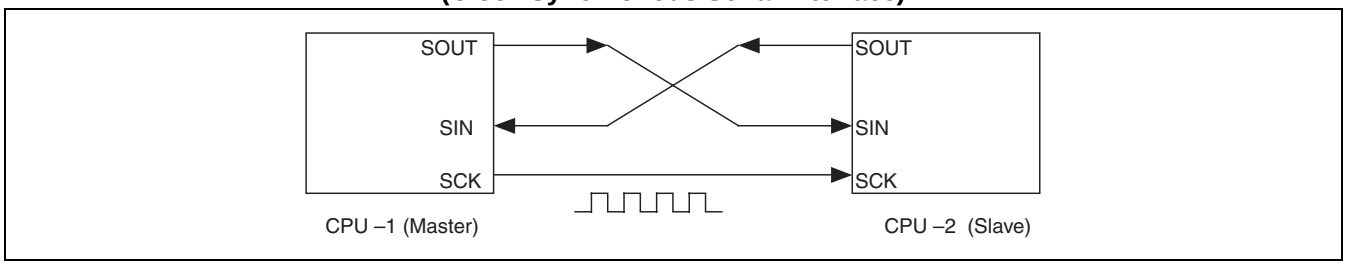
28.16 Setup Procedure and Program Flow for CSIO (Clock Synchronous Serial Interface)

Two-way serial synchronous communication is enabled in CSIO (clock synchronous serial interface).

■ Connection Between CPUs

Two-way communication should be selected for CSIO (clock synchronous serial interface). Two CPUs are connected to each other, as shown in Figure 28.16-1.

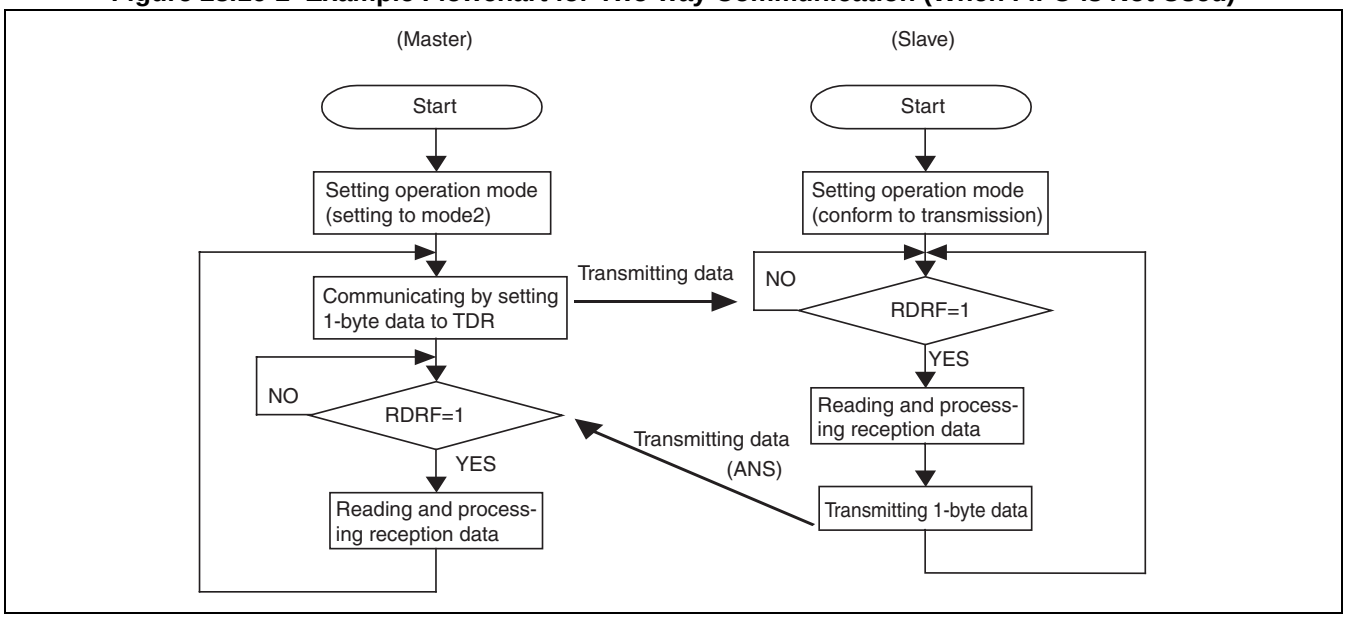
Figure 28.16-1 Example of Two-way Communication Connection for CSIO (Clock Synchronous Serial Interface)



■ Flowchart

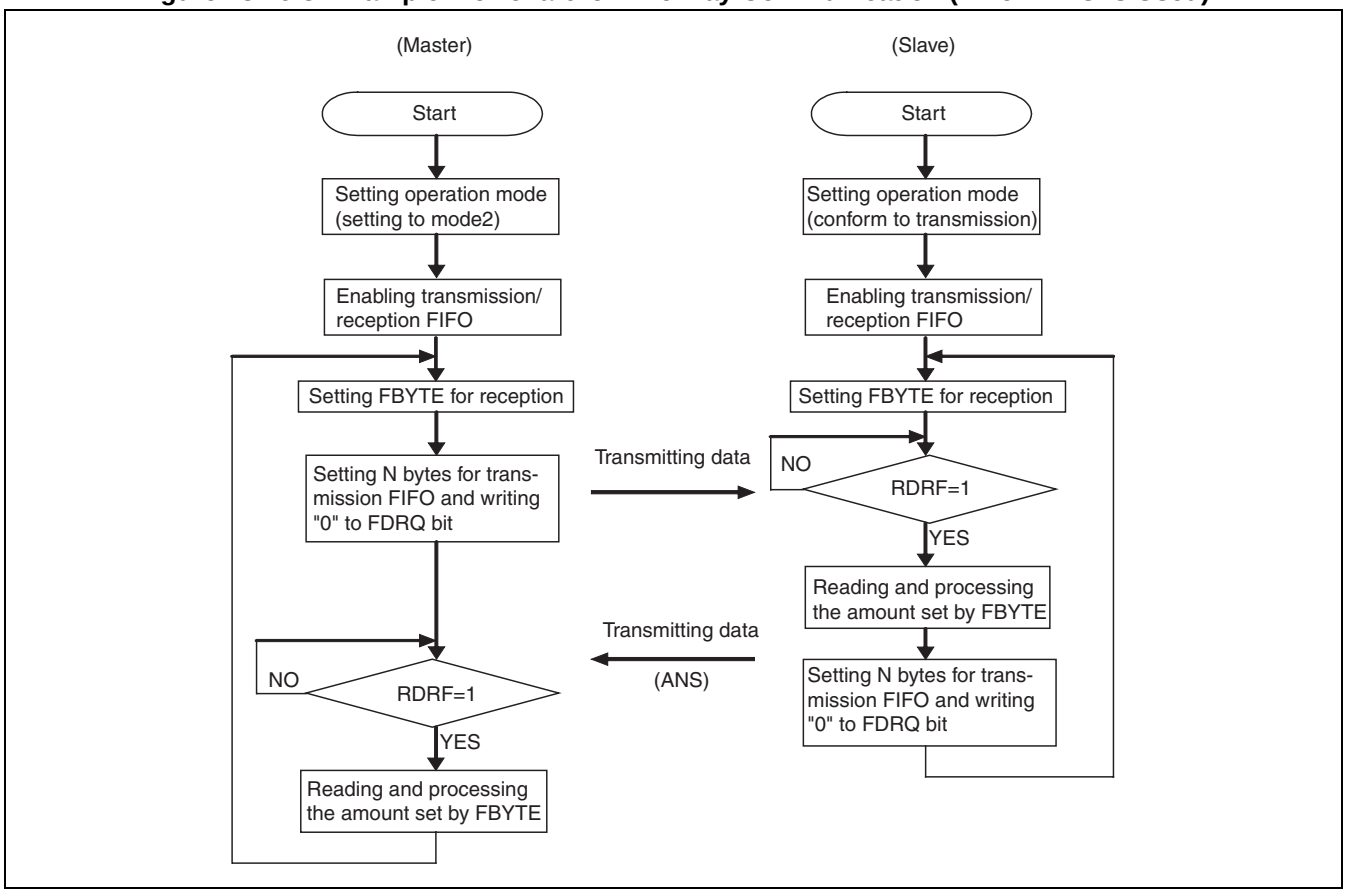
● When FIFO is not used

Figure 28.16-2 Example Flowchart for Two-way Communication (When FIFO is Not Used)



● When FIFO is used

Figure 28.16-3 Example Flowchart for Two-way Communication (When FIFO is Used)



28.17 I²C Interface

Of all the functions of the multi-function serial interface, this section describes the I²C interface that is supported in operation mode 4.

- I²C Interface
- Overview of I²C Interface
- Registers of I²C Interface
  - I²C Bus Control Register (IBCR)
  - Serial Mode Register (SMR)
  - I²C Bus Status Register (IBSR)
  - Serial Status Register (SSR)
  - Reception Data Register / Transmission Data Register (RDR/TDR)
  - 7-bit Slave Address Mask Register (ISMK)
  - 7-bit Slave Address Register (ISBA)
  - Baud Rate Generator Registers 1, 0 (BGR1, BGR0)
  - FIFO Control Register 1 (FCR1)
  - FIFO Control Register 0 (FCR0)
  - FIFO Byte Register (FBYTE1/FBYTE2)
- Interrupts of I²C Interface
  - Operation of I²C Interface Communication
  - Master Mode
  - Slave Mode
  - Bus Error
- Dedicated Baud Rate Generator
  - Example Flowchart for I²C Interface

28.18 Overview of I²C Interface

The I²C interface supports a bus between ICs and operates as a master/slave device on the I²C bus. This interface also comes with transmission/reception FIFO (up to 16 bytes each). There is no I²C function for ch.0.

■ Functions of I²C Interface

The I²C interface has the following functions.

- Master/slave transmission & reception functionality
- Arbitration function
- Clock synchronization
- Transmission direction detection
- Generation and detection of repeated start condition
- Bus error detection
- General call addressing
- 7-bit addressing as master/slave
- Interrupts can be generated during transmission and bus errors.
- 10-bit addressing can be supported by a program.

■ Functions of FIFO

The FIFO has the following functions.

- Transmission/reception FIFO mounted (maximum capacity: transmission FIFO = 16 bytes; reception FIFO = 16 bytes)*
- Transmission FIFO or reception FIFO selectable
- Transmission data can be resent.
- The interrupt timing for reception FIFO can be modified by software.
- FIFO reset is supported separately.

*: There is no FIFO between ch.0 and ch.7.

28.19 Registers of I²C Interface

This section lists the registers of the I²C interface.

■ List of Registers of I²C Interface

Table 28.19-1Registers of the I²C (1 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
1	IBCR1	I²C bus control register 1	28.19.1
	SMR1	Serial mode register 1	28.19.2
	IBSR1	I²C bus status register 1	28.19.3
	BGR1	Baud rate generator register 1	28.19.8
	SSR1	Serial status register 1	28.19.4
	RDR1	Received data register 1	28.19.5
	TDR1	Transmitted data register 1	28.19.5
	ISMK1	7-bit slave address mask register 1	28.19.6
	ISBA1	7-bit slave address register 1	28.19.7
2	IBCR2	I²C bus control register 2	28.19.1
	SMR2	Serial mode register 2	28.19.2
	IBSR2	I²C bus status register 2	28.19.3
	BGR2	Baud rate generator register 2	28.19.8
	SSR2	Serial status register 2	28.19.4
	RDR2	Received data register 2	28.19.5
	TDR2	Transmitted data register 2	28.19.5
	ISMK2	7-bit slave address mask register 2	28.19.6
	ISBA2	7-bit slave address register 2	28.19.7

Table 28.19-1Registers of the I²C (2 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
3	IBCR3	I²C bus control register 3	28.19.1
	SMR3	Serial mode register 3	28.19.2
	IBSR3	I²C bus status register 3	28.19.3
	BGR3	Baud rate generator register 3	28.19.8
	SSR3	Serial status register 3	28.19.4
	RDR3	Received data register 3	28.19.5
	TDR3	Transmitted data register 3	28.19.5
	ISMK3	7-bit slave address mask register 3	28.19.6
	ISBA3	7-bit slave address register 3	28.19.7
4	IBCR4	I²C bus control register 4	28.19.1
	SMR4	Serial mode register 4	28.19.2
	IBSR4	I²C bus status register 4	28.19.3
	BGR4	Baud rate generator register 4	28.19.8
	SSR4	Serial status register 4	28.19.4
	RDR4	Received data register 4	28.19.5
	TDR4	Transmitted data register 4	28.19.5
	ISMK4	7-bit slave address mask register 4	28.19.6
	ISBA4	7-bit slave address register 4	28.19.7
5	IBCR5	I²C bus control register 5	28.19.1
	SMR5	Serial mode register 5	28.19.2
	IBSR5	I²C bus status register 5	28.19.3
	BGR5	Baud rate generator register 5	28.19.8
	SSR5	Serial status register 5	28.19.4
	RDR5	Received data register 5	28.19.5
	TDR5	Transmitted data register 5	28.19.5
	ISMK5	7-bit slave address mask register 5	28.19.6
	ISBA5	7-bit slave address register 5	28.19.7

Table 28.19-1Registers of the I^2C (3 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
6	IBCR6	I^2C bus control register 6	28.19.1
	SMR6	Serial mode register 6	28.19.2
	IBSR6	I^2C bus status register 6	28.19.3
	BGR6	Baud rate generator register 6	28.19.8
	SSR6	Serial status register 6	28.19.4
	RDR6	Received data register 6	28.19.5
	TDR6	Transmitted data register 6	28.19.5
	ISMK6	7-bit slave address mask register 6	28.19.6
	ISBA6	7-bit slave address register 6	28.19.7
7	IBCR7	I^2C bus control register 7	28.19.1
	SMR7	Serial mode register 7	28.19.2
	IBSR7	I^2C bus status register 7	28.19.3
	BGR7	Baud rate generator register 7	28.19.8
	SSR7	Serial status register 7	28.19.4
	RDR7	Received data register 7	28.19.5
	TDR7	Transmitted data register 7	28.19.5
	ISMK7	7-bit slave address mask register 7	28.19.6
	ISBA7	7-bit slave address register 7	28.19.7

Table 28.19-1Registers of the I²C (4 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
8	IBCR8	I²C bus control register 8	28.19.1
	SMR8	Serial mode register 8	28.19.2
	IBSR8	I²C bus status register 8	28.19.3
	BGR8	Baud rate generator register 8	28.19.8
	SSR8	Serial status register 8	28.19.4
	RDR8	Received data register 8	28.19.5
	TDR8	Transmitted data register 8	28.19.5
	FCR18	FIFO control register 18	28.19.9
	FCR08	FIFO control register 08	28.19.10
	FBYTE18	FIFO1 byte register 8	28.19.10
	FBYTE28	FIFO2 byte register 8	28.19.11
	ISMK8	7-bit slave address mask register 8	28.19.6
	ISBA8	7-bit slave address register 8	28.19.7
9	IBCR9	I²C bus control register 9	28.19.1
	SMR9	Serial mode register 9	28.19.2
	IBSR9	I²C bus status register 9	28.19.3
	BGR9	Baud rate generator register 9	28.19.8
	SSR9	Serial status register 9	28.19.4
	RDR9	Received data register 9	28.19.5
	TDR9	Transmitted data register 9	28.19.5
	FCR19	FIFO control register 19	28.19.9
	FCR09	FIFO control register 09	28.19.10
	FBYTE19	FIFO1 byte register 9	28.19.10
	FBYTE29	FIFO2 byte register 9	28.19.11
	ISMK9	7-bit slave address mask register 9	28.19.6
	ISBA9	7-bit slave address register 9	28.19.7

Table 28.19-1Registers of the I²C (5 / 5)

Channel	Abbreviated Register Name	Register Name	Reference
10	IBCR10	I²C bus control register 10	28.19.1
	SMR10	Serial mode register 10	28.19.2
	IBSR10	I²C bus status register 10	28.19.3
	BGR10	Baud rate generator register 10	28.19.8
	SSR10	Serial status register 10	28.19.4
	RDR10	Received data register 10	28.19.5
	TDR10	Transmitted data register 10	28.19.5
	FCR110	FIFO control register 110	28.19.9
	FCR010	FIFO control register 010	28.19.10
	FBYTE110	FIFO1 byte register 10	28.19.10
	FBYTE210	FIFO2 byte register 10	28.19.11
	ISMK10	7-bit slave address mask register 10	28.19.6
	ISBA10	7-bit slave address register 10	28.19.7
11	IBCR11	I²C bus control register 11	28.19.1
	SMR11	Serial mode register 11	28.19.2
	IBSR11	I²C bus status register 11	28.19.3
	BGR11	Baud rate generator register 11	28.19.8
	SSR11	Serial status register 11	28.19.4
	RDR11	Received data register 11	28.19.5
	TDR11	Transmitted data register 11	28.19.5
	FCR111	FIFO control register 111	28.19.9
	FCR011	FIFO control register 011	28.19.10
	FBYTE111	FIFO1 byte register 11	28.19.10
	FBYTE211	FIFO2 byte register 11	28.19.11
	ISMK11	7-bit slave address mask register 11	28.19.6
	ISBA11	7-bit slave address register 11	28.19.7

Table 28.19-2 Bit Assignment of I^2C Interface

	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
IBCR/ SMR	MSS	ACT/ SCC	ACKE	WSEL	CNDE	INTE	BER	INT	MD2	MD1	MD0	-	RIE	TIE	-	-
SSR/IBSR	REC	TSET	-	-	ORE	RDRF	TDRE	-	FBT	RACK	RSA	TRX	AL	RSC	SPC	BB
RDR/TDR	-	-	-	-	-	-	-	-	D7	D6	D5	D4	D3	D2	D1	D0
BGR1/ BGR0	-	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
ISMK/ ISBA	EN	SM6	SM5	SM4	SM3	SM2	SM1	SM0	SAEN	SA6	SA5	SA4	SA3	SA2	SA1	SA0
FCR1/ FCR0	-	-	-	FLSTE	FRIIE	FDRQ	FTIE	FSEL	-	FLST	FLD	FSET	FCL2	FCL1	FE2	FE1
FBYTE2/ FBYTE1	FD15	FD14	FD13	FD12	FD11	FD10	FD9	FD8	FD7	FD6	FD5	FD4	FD3	FD2	FD1	FD0

28.19.1 I²C Bus Control Register (IBCR)

The I²C bus control register (IBCR) selects master/slave mode, generates a repeated start condition, enables the acknowledge function, enables interrupts, bus error detection and displays an interrupt flag.

■ I²C Bus Control Register (IBCR)

Figure 28.19-1 shows the bit structure of the I²C bus control register (IBCR), and Table 28.19-3 describes the function of each bit.

Figure 28.19-1 Bit Structure of I²C Bus Control Register (IBCR)

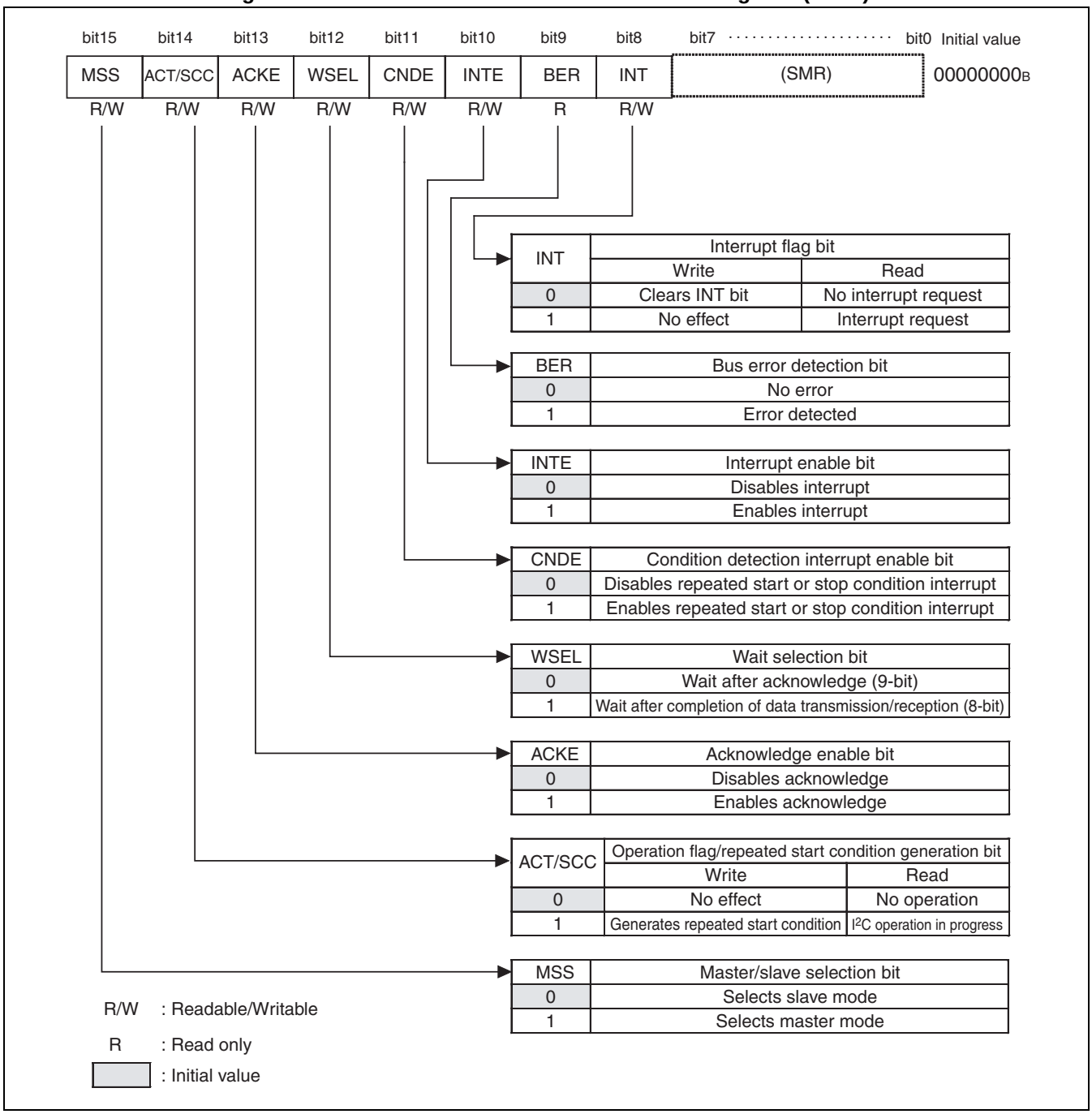

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ····· bit0 Initial value
MSS ACT/SCC ACKE WSEL CNDE INTE BER INT (SMR) 00000000B
R/W R/W R/W R/W R/W R/W R R/W
INT Interrupt flag bit
Write Read
0 Clears INT bit No interrupt request
1 No effect Interrupt request
BER Bus error detection bit
0 No error
1 Error detected
INTE Interrupt enable bit
0 Disables interrupt
1 Enables interrupt
CNDE Condition detection interrupt enable bit
0 Disables repeated start or stop condition interrupt
1 Enables repeated start or stop condition interrupt
WSEL Wait selection bit
0 Wait after acknowledge (9-bit)
1 Wait after completion of data transmission/reception (8-bit)
ACKE Acknowledge enable bit
0 Disables acknowledge
1 Enables acknowledge
ACT/SCC Operation flag/repeated start condition generation bit
Write Read
0 No effect No operation
1 Generates repeated start condition I²C operation in progress
MSS Master/slave selection bit
0 Selects slave mode
1 Selects master mode
R/W : Readable/Writable
R : Read only
: Initial value

Table 28.19-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (1 / 4)

<table>
<tr><th colspan="2">Bit name</th><th>Function</th></tr>
<tr><td>bit15</td><td>MSS:
Master/slave selection bit</td><td>
• Master mode will be selected if this bit is set to "1" when the I²C bus is in the idle state (EN = 1, BB = 0).
• If this bit is set to "1" when the BB bit in the IBSR register is set to "1", the register will wait to generate a start condition until the BB bit becomes "0". If a slave address match occurs during that wait and the device operates as a slave, this bit will be set to "0" and the AL bit in the IBSR register will be set to "1".
• A stop condition will be generated if "0" is written to this bit when the device is operating as the master (MSS = 1, ACT = 1) and the interrupt flag (INT) is set to "1".
The MSS bit is cleared under the following conditions.
• The I²C interface is disabled (EN bit = 0).
• An arbitration lost condition occurs.
• A bus error is detected (BER bit = 1).
• "0" is written to the MSS bit when INT is "1".
The relationship between the MSS and ACT bits is shown below.
<table>
<tr><th>MSS bit</th><th>ACT bit</th><th>Status</th></tr>
<tr><td>0</td><td>0</td><td>Idle</td></tr>
<tr><td>0</td><td>1</td><td>Slave operation in progress (slave mode) due to slave address match or ACK response* to reserved address</td></tr>
<tr><td>1</td><td>0</td><td>Master operation on standby</td></tr>
<tr><td>1</td><td>1</td><td>Master operation in progress (master mode)</td></tr>
</table>
*: ACK response: indicates that SDA of the I²C bus is at "L" during acknowledge.
Note:
Change the MSS bit from "1" to "0" when the INT bit and MSS bit are set to "1". If "0" is written to the MSS bit when the ACT bit is set to "1", the INT bit will also be cleared to "0".
Writing "0" to the MSS bit returns "1" during master operation, as long as the ACT bit is set to "1".
</td></tr>
</table>

Table 28.19-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (2 / 4)

Bit name		Function
bit14	ACT/SCC: Operation flag / repeated start condition generation bit	This bit has different meanings between read and write. **Read**: ACT bit **Write**: SCC bit The ACT bit indicates that the device is operating in master or slave mode. Setting conditions for the ACT bit: • A start condition is output to the I²C bus (master mode). • A slave address matches the address transmitted from the master (slave mode). • A reserved address is detected and then an acknowledge is returned as a response (MSS = 0: slave mode). Reset conditions for the ACT bit: <Master mode> • A stop condition is detected. • An arbitration lost condition is detected. • A bus error is detected. • The I²C interface is disabled (EN bit = 0). <Slave mode> • A (repeated) start condition is detected. • A stop condition is detected. • An acknowledge is not returned although a reserved address is detected (RSA bit = 1). • The I²C interface is disabled (EN bit = 0). • A bus error occurs (BER bit = 1). A repeated start is performed when "1" is written to this bit during master mode. Writing "0" is invalid. Note: Write "1" to the SCC bit while an interrupt is occurring in master mode (MSS = 1, ACT = 1, INT = 1). The INT bit will be cleared to "0" if "1" is written to the SCC bit when the ACT bit is set to "1". In slave mode (MSS = 0, ACT = 1), it is prohibited to write "1" to this bit. The MSS bit has higher priority than the SCC bit, when "1" is written to the SCC bit and "0" is written to the MSS bit. The SCC bit is read when a read modify write (RMW) instruction is used.

Table 28.19-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (3 / 4)

Bit name		Function
bit13	ACKE: Acknowledge enable bit	• If this bit is set to "1", "L" will be output when an acknowledge is returned. • When ACT is set to "1", this bit must be modified, if necessary, while the INT bit is set to "1". This bit is invalid under the following conditions. • An acknowledge is returned to an address field other than the reserved address (automatic generation). • Data transmission (RSA = 0, TRX = 1, FBT = 0) • An ACK is returned whenever the reception FIFO is enabled and slave reception is selected (FE = 1, MSS = 0, ACT = 1). • When the reception FIFO is enabled, WSEL is "0", and master reception is selected (FE = 1, MSS = 1, ACT = 1, WSEL = 0), setting the TDRE bit to "0" returns an ACK while setting it to "1" returns a NACK. An ACK is always returned when the reception FIFO is enabled, WSEL is "0", and slave transmission is performed through reserved address detection (RSA = 1, TRX = 1, FBT = 1). To allow a NACK to be returned, disable the reception FIFO and set ACKE to "0" during an interrupt after the reserved address detection. • The reception FIFO is enabled, WSEL is "1", and the transmission data register contains data in master reception (FE = 1, MSS = 1, ACT = 1, WSEL = 1, TDRE = 0).

bit12	WSEL:
	Wait selection bit
		• This bit is used to determine whether an interrupt should occur (INT = 1) before or after an acknowledgement to put the I²C bus in a wait state.
		• The WSEL bit is invalid under the following conditions.
		  • An interrupt occurs for the first byte*1 (INT = 1).
		  • A reserved address is detected (FBT = 1, RSA = 1).
		  • A NACK response*2 is detected during a data transfer when FIFO is used (FE = 1, RACK = 1, ACT = 1).
		  • The reception FIFO becomes full when it is used.
		*1: First byte: indicates the data after a (repeated) start condition
		*2: NACK response: indicates that SDA of the I²C bus is at "H" during acknowledgement.

bit11	CNDE: Condition detection interrupt enable bit	This bit is used to enable the occurrence of interrupts when a stop condition or a repeated start condition is detected in master or slave mode (ACT = 1). An interrupt occurs when the RSC or SPC bit in the IBSR register is set to "1" and this bit is set to "1".
bit10	INTE: Interrupt enable bit	This bit is used to enable an interrupt (INT = 1) for data transmission/reception and a bus error in master or slave mode.
bit9	BER: Bus error detection bit	This bit indicates that an error is detected on the I²C bus. Setting conditions for the BER bit: • A start condition or stop condition is detected during the transfer of the first byte*. • A (repeated) start condition or stop condition is detected at the 2nd bit - 9th (acknowledge) bit of data in the second or succeeding byte. Reset conditions for the BER bit: • "0" is written to the INT bit when BER is set to "1". • The I²C interface is disabled (EN = 0). *: First byte: indicates the data after (repeated) start condition Note: Data cannot be transmitted or received properly if this bit is set to "1" when the interrupt flag (INT bit) is set to "1". In this case, take an action such as retransmitting the data.

Table 28.19-3 Functional Description of Each Bit of I²C Bus Control Register (IBCR) (4 / 4)

Bit name		Function
bit8	INT: Interrupt flag bit	This bit is set to "1" after the 8th or 9th bit (ACK) of data transmission/reception in master or slave mode, or upon the occurrence of a bus error. In cases other than the occurrence of a bus error, SCL is set to "L" when the INT bit is set to "1". When the INT bit is set to "0", SCL is released from the "L" state. Setting conditions for the INT bit: <8th bit> • A reserved address is detected in the first byte. • WSEL is "1", and an arbitration lost condition is detected in the second or succeeding byte. • WSEL is "1", and the TDRE bit is set to "1" in the second or succeeding byte during master operation. • WSEL is "1", the reception FIFO is disabled and the TDRE bit is set to "1" in the second or succeeding byte during slave operation. • WSEL is set to "1", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. <9th bit> • An arbitration lost condition is detected in the first byte. • A NACK is received at times other than when a stop condition output is set ("0" written to the MSS bit during master operation). • The TDRE bit is set to "1" in the transmission direction (TRX = 1) of master or slave mode without the detection of a reserved address in the first byte. • The reception FIFO contains data when the reception FIFO is enabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. • The TDRE bit is set to "1" when the reception FIFO is disabled in the reception direction (TRX = 0) of master or slave mode without the detection of a reserved address in the first byte. • WSEL is set to "0", and an arbitration lost condition is detected in the second or succeeding byte. • WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during master mode operation. • WSEL is set to "0", and the TDRE bit is set to "1" in the second or succeeding byte during slave transmission. • WSEL is set to "0", the reception FIFO is disabled, and slave reception is selected. In slave reception, however, an interrupt does not occur in the 9th bit of the first byte in which a reserved address is detected. • The reception FIFO is enabled, and it becomes full in slave reception. <Other condition> A bus error is detected. Reset conditions for the INT bit: • (1) "0" is written to the INT bit. • (2) "0" is written to the MSS bit when the INT bit is "1" and the ACT bit is "1". • (3) "1" is written to the SCC bit when the INT bit is "1" and the ACT bit is "1". Writing "1" to the INT bit is invalid. Note: Setting the EN bit to "0" may set the RDRF and INT bits to "1", depending on the reception timing. In this case, read the reception data to clear the INT bit. "1" is read when a read modify write (RMW) instruction is used. The INT bit cannot be set to "1" even if the reception FIFO is full in master reception operation when the reception FIFO is enabled.

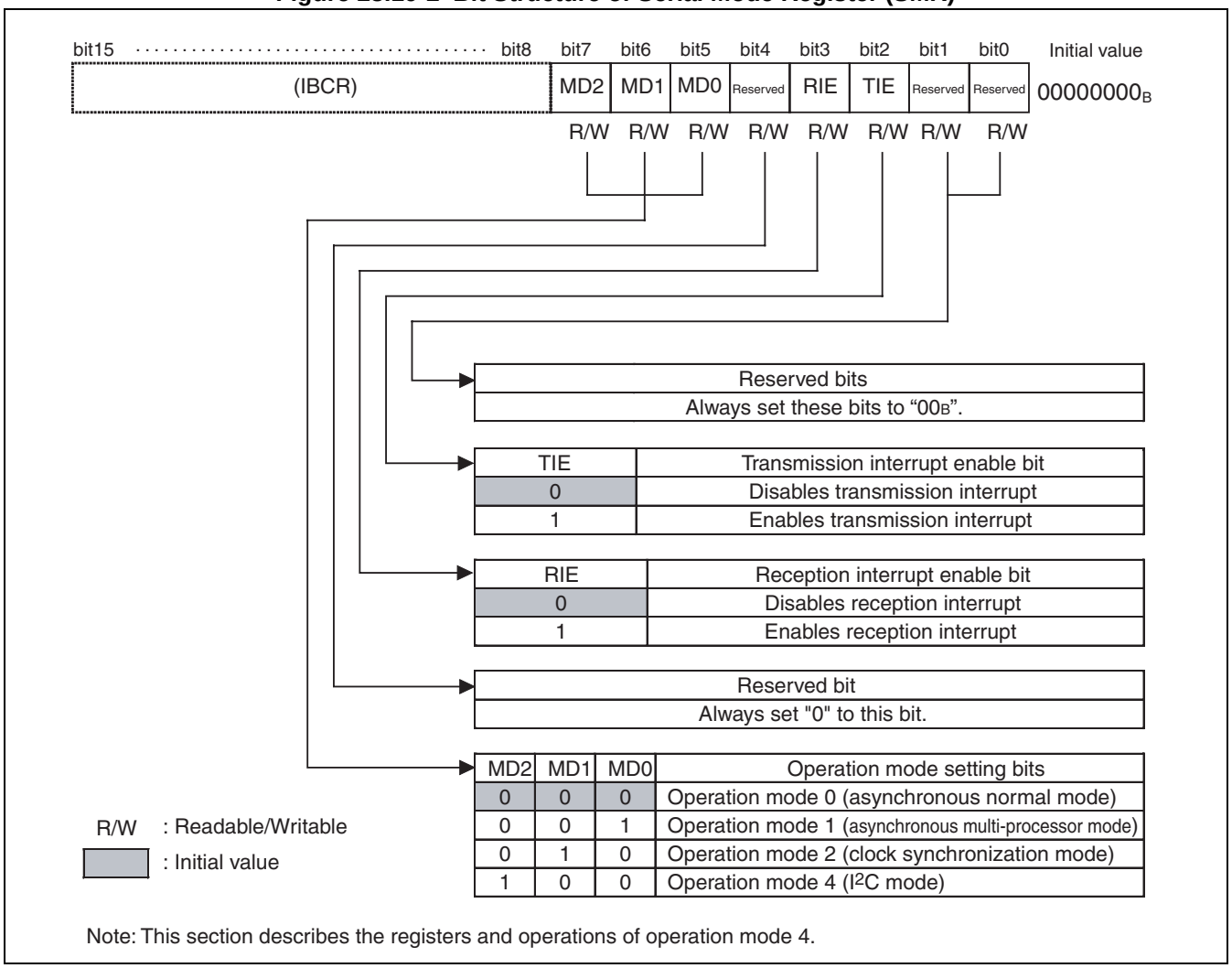
28.19.2 Serial Mode Register (SMR)

The serial mode register (SMR) sets the operation mode, and enables or disables transmission/ reception interrupts.

■ Serial Mode Register (SMR)

Figure 28.19-2 shows the bit structure of the serial mode register (SMR), and Table 28.19-4 describes the function of each bit.

Figure 28.19-2 Bit Structure of Serial Mode Register (SMR)


bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(IBCR) MD2 MD1 MD0 Reserved RIE TIE Reserved Reserved 00000000B
R/W R/W R/W R/W R/W R/W R/W R/W
Reserved bits
Always set these bits to "00B".
TIE Transmission interrupt enable bit
0 Disables transmission interrupt
1 Enables transmission interrupt
RIE Reception interrupt enable bit
0 Disables reception interrupt
1 Enables reception interrupt
Reserved bit
Always set "0" to this bit.
MD2 MD1 MD0 Operation mode setting bits
0 0 0 Operation mode 0 (asynchronous normal mode)
0 0 1 Operation mode 1 (asynchronous multi-processor mode)
0 1 0 Operation mode 2 (clock synchronization mode)
1 0 0 Operation mode 4 (I2C mode)
R/W : Readable/Writable
: Initial value

Note: This section describes the registers and operations of operation mode 4.

Table 28.19-4 Functional Description of Each Bit of Serial Mode Register (SMR)

Bit name		Function
bit7 to bit5	MD2 to MD0: Operation mode setting bits	These bits are used to select the operation mode. "000_B": Selects operation mode 0 (asynchronous normal mode) "001_B": Selects operation mode 1 (asynchronous multi-processor mode) "010_B": Selects operation mode 2 (clock synchronization mode) "100_B": Selects operation mode 4 (I^2C mode) This section describes the registers and operations of operation mode 4 (I^2C mode). Note: Settings other than above are prohibited. To switch the operation mode, disable I^2C first (ISMK:EN = 0). Set each register after selecting the operation mode.
bit4	Reserved bit	Always set "0" to this bit.
bit3	RIE: Reception interrupt enable bit	• This bit is used to enable or disable the output of reception interrupt requests to the CPU. • A reception interrupt request is output when the RIE bit and the reception data flag bit (RDRF) are set to "1", or the error flag bit (ORE) is set to "1". Note: Set this bit to "0" when receiving data using the INT bit in the I^2C bus control register (IBCR).
bit2	TIE: Transmission interrupt enable bit	• This bit is used to enable or disable the output of transmission interrupt requests to the CPU. • A transmission interrupt request is output when the TIE and TDRE bits are set to "1". Note: Set this bit to "0" when transmitting data using the INT bit in the I^2C bus control register (IBCR).
bit1, bit0	Reserved bits	Always set these bits to "00_B".

<Note>

The operation mode must be set first. Otherwise, the other registers will be initialized when the operation mode is changed. Note, however, that when IBCR and SMR are written simultaneously with 16-bit write access, IBCR reflects the written content.

28.19.3 I²C Bus Status Register (IBSR)

The I²C bus status register (IBSR) indicates the detection of a first byte, a reserved address, a repeated start condition, acknowledge, data direction, arbitration lost condition, stop condition, and I²C bus status.

■ I²C Bus Status Register (IBSR)

Figure 28.19-3 shows the bit structure of the I²C bus status register (IBSR) and Table 28.19-5 describes the function of each bit.

Figure 28.19-3 Bit Structure of I²C Bus Status Register (IBSR)

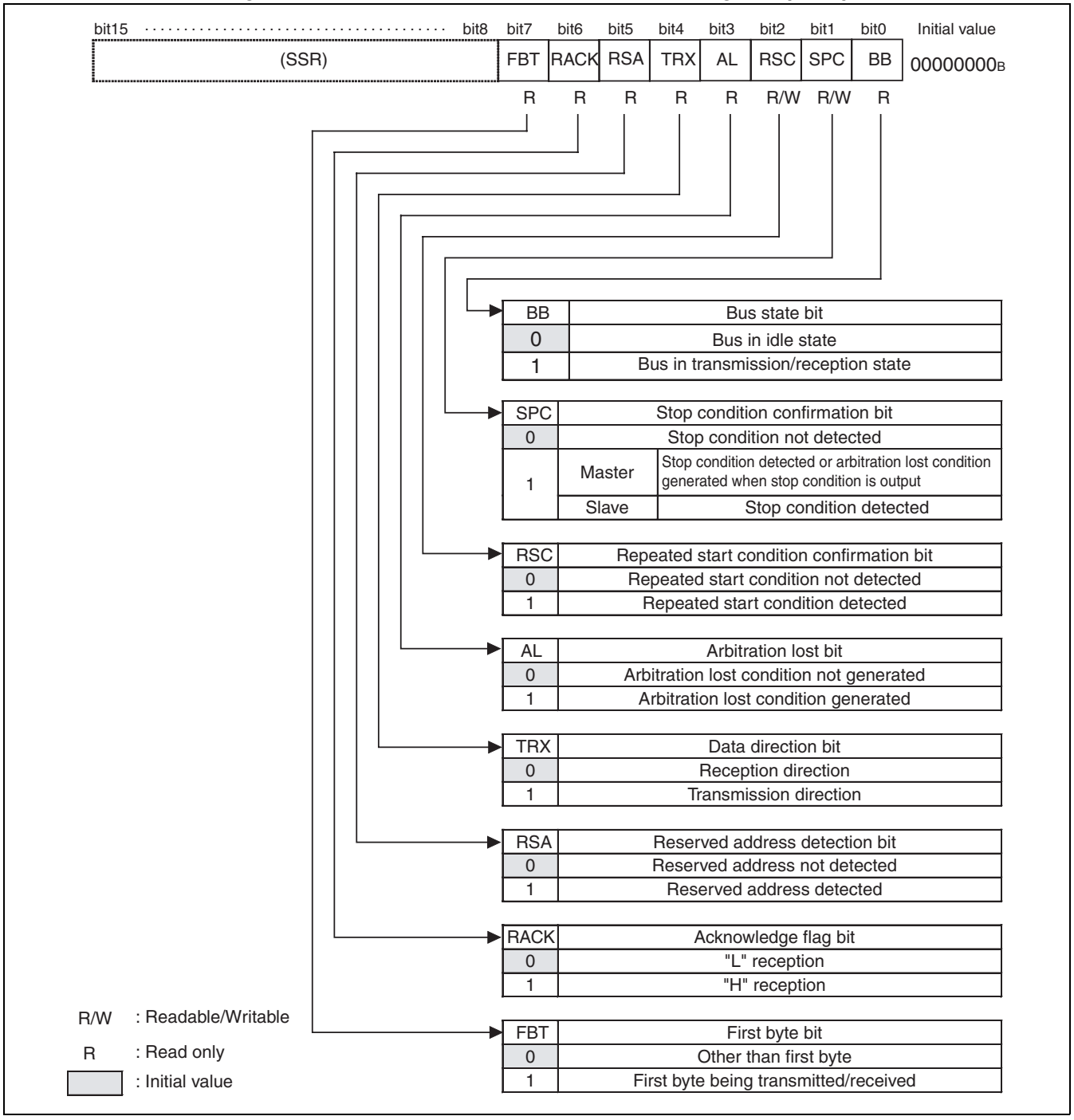

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(SSR) FBT RACK RSA TRX AL RSC SPC BB 00000000B
R R R R R R/W R/W R
BB Bus state bit
0 Bus in idle state
1 Bus in transmission/reception state
SPC Stop condition confirmation bit
0 Stop condition not detected
1 Master Stop condition detected or arbitration lost condition generated when stop condition is output
Slave Stop condition detected
RSC Repeated start condition confirmation bit
0 Repeated start condition not detected
1 Repeated start condition detected
AL Arbitration lost bit
0 Arbitration lost condition not generated
1 Arbitration lost condition generated
TRX Data direction bit
0 Reception direction
1 Transmission direction
RSA Reserved address detection bit
0 Reserved address not detected
1 Reserved address detected
RACK Acknowledge flag bit
0 "L" reception
1 "H" reception
FBT First byte bit
0 Other than first byte
1 First byte being transmitted/received
R/W : Readable/Writable
R : Read only
: Initial value

Table 28.19-5 Functional Description of Each Bit of I²C Bus Status Register (1 / 3)

Bit name		Function
bit7	FBT: First byte bit	This bit indicates the first byte. Setting condition for the FBT bit: A (repeated) start condition is detected. Clearing conditions for the FBT bit: (1) The second byte is transmitted or received. (2) A stop condition is detected. (3) The I²C interface is disabled (EN bit = 0). (4) A bus error is detected (BER bit = 1).
bit6	RACK: Acknowledge flag bit	This bit is used to indicate the acknowledge received for the first byte during master or slave mode. Update conditions for the RACK bit (1) Acknowledge for the first byte (2) Acknowledge for data in master or slave mode Clearing conditions for the RACK bit (RACK bit = 0) (1) A (repeated) start condition is detected. (2) The I²C interface is disabled (EN bit = 0). (3) A bus error is detected (BER bit = 1).
bit5	RSA: Reserved address detection bit	This bit indicates the detection of a reserved address. Setting condition for the RSA bit (RSA = 1) The first byte is set to "0000XXXX" or "1111XXXX". "X" can be "0" or "1". Reset conditions for the RSA bit (RSA = 0) (1) A (repeated) start condition is detected. (2) A stop condition is detected. (3) The I²C interface is disabled (EN bit = 0). (4) A bus error is detected (BER bit = 1). When the RSA bit is set to "1" in the first byte, the interrupt flag (INT) is set to "1" at the falling edge of SCL in the 8th bit of the first byte to set the SCL to "L", whether the FIFO is enabled or disabled. In this case, ACKE should be set to "1" and the interrupt flag (INT) should be cleared to "0" in order to read reception data and allow the device to operate as a slave. If the TRX bit is set to "0", the device will receive data as a slave. To disable data reception in the middle of the operation, set the ACKE bit to "0". No more data will be received afterward. Note: When ACKE is set to "0" during a data transfer, it is prohibited to set ACKE to "1" until a stop condition or repeated start condition is detected. If slave transmission is confirmed during an interrupt by the detection of a reserved address, an ACK will be returned when the reception FIFO has been enabled. Therefore, disable the reception FIFO and set ACKE to "0".

Table 28.19-5 Functional Description of Each Bit of I^2C Bus Status Register (2 / 3)

Bit name		Function
bit4	TRX: Data direction bit	This bit indicates the data direction. Setting conditions for the TRX bit: (1) A (repeated) start condition is transmitted in master mode. (2) The 8th bit of the first byte is "1" in slave mode (transmission direction as a slave). Reset conditions for the TRX bit: (1) An arbitration lost condition is generated (AL = 1). (2) The 8th bit of the first byte is "0" in slave mode (reception direction as a slave). (3) The 8th bit of the first byte is "1" in master mode (reception direction as the master). (4) A stop condition is detected. (5) A (repeated) start condition is detected in modes other than master mode. (6) The I^2C interface is disabled (EN bit = 0). (7) A bus error is detected (BER bit = 1).
bit3	AL: Arbitration lost bit	This bit indicates an arbitration lost condition. Setting conditions for the AL bit: (1) The output data is different from the received data in master mode. (2) The device is operating as a slave although the MSS bit has been set to "1". (3) A repeated start condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. (4) A stop condition is detected in the first bit of the data contained in the second or succeeding byte in master mode. (5) A repeated start condition cannot be generated in master mode, despite attempts to do so. (6) A stop condition cannot be generated in master mode, despite attempts to do so. Reset conditions for the AL bit: (1) "1" is written to the MSS bit. (2) "0" is written to the INT bit. (3) "0" is written to the SPC bit when the AL and SPC bits are set to "1". (4) The I^2C interface is disabled (EN bit = 0). (5) A bus error is detected (BER bit = 1).

Table 28.19-5 Functional Description of Each Bit of I^2C Bus Status Register (3 / 3)

Bit name		Function
bit2	RSC: Repeated start condition confirmation bit	This bit indicates that a repeated start condition has been detected in master or slave mode. Setting condition for the RSC bit: A repeated start condition is detected after acknowledgement during slave or master mode operation. Reset conditions for the RSC bit: (1) "0" is written to the RSC bit. (2) "1" is written to the MSS bit. (3) The I^2C interface is disabled (EN bit = 0). Writing "1" to this bit is invalid. Note: Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a repeated start condition is detected. "1" is read when a read modify write (RMW) instruction is used.
bit1	SPC: Stop condition confirmation bit	This bit indicates that a stop condition has been detected in master or slave mode. Setting conditions for the SPC bit: (1) A stop condition is detected during slave or master mode operation. (2) An arbitration lost condition is generated when a stop condition is generated in master mode. Reset conditions for the SPC bit: (1) "0" is written to this bit. (2) "1" is written to the MSS bit. (3) The I^2C interface is disabled (EN bit = 0). Writing "1" to this bit is invalid. Note: Slave mode will end unless ACK is returned when the device is operating reception in slave mode by the detection of a reserved address. Consequently, this bit will not be set to "1" even if a stop condition is detected. "1" is read when a read modify write (RMW) instruction is used.
bit0	BB: Bus state bit	This bit indicates the bus state. Setting condition for the BB bit: "L" is detected at SDA or SCL of the I^2C bus. Reset conditions for the BB bit: (1) A stop condition is detected. (2) The I^2C interface is disabled (EN bit = 0). (3) A bus error is detected (BER bit = 1).

28.19.4 Serial Status Register (SSR)

The serial status register (SSR) checks the transmission/reception status.

■ Serial Status Register (SSR)

Figure 28.19-4 shows the bit structure of the serial status register (SSR) and Table 28.19-6 describes the function of each bit.

Figure 28.19-4 Bit Structure of Serial Status Register (SSR)

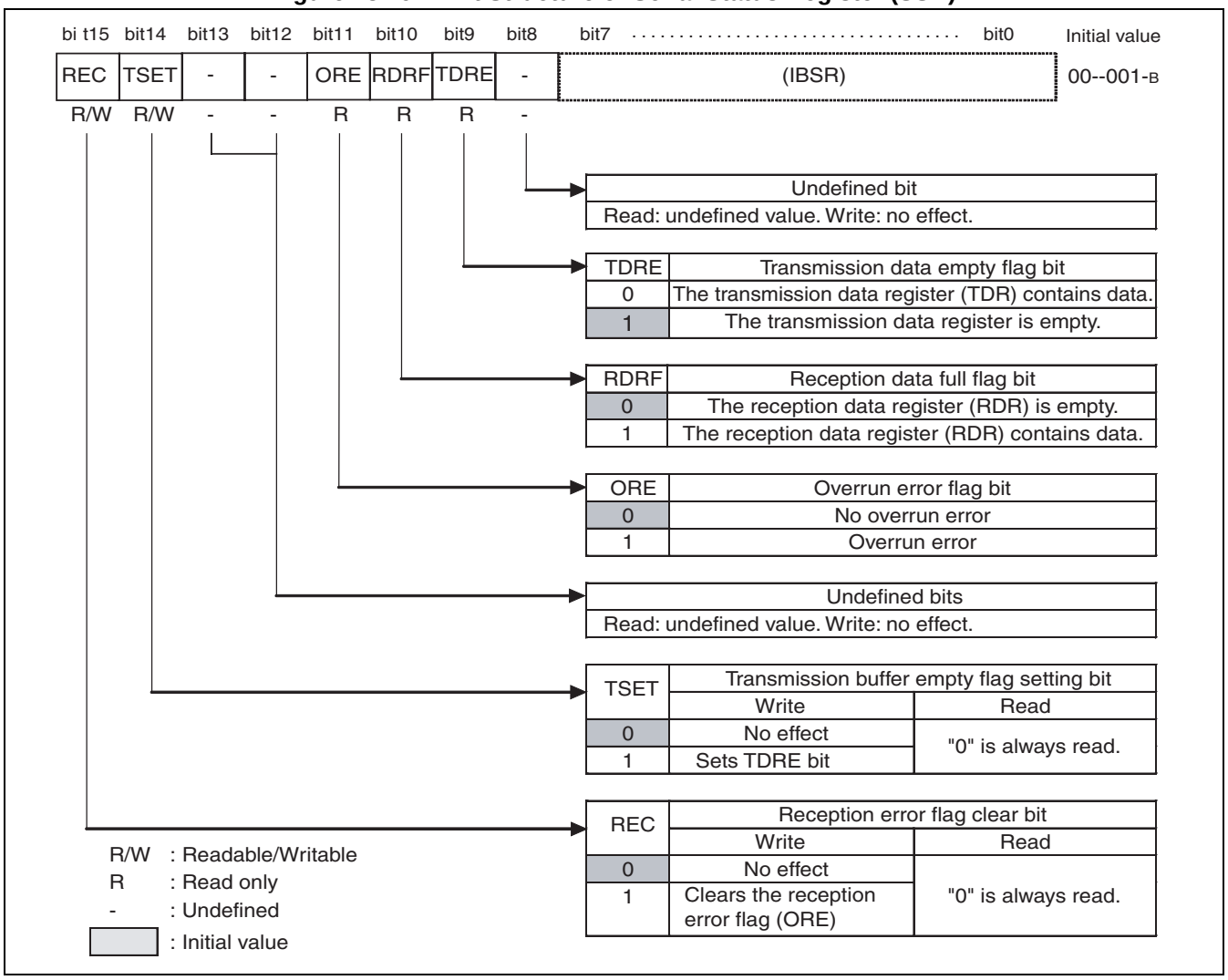

bi t15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
REC TSET - - ORE RDRF TDRE - (IBSR) 00--001-B
R/W R/W - - R R R -
Undefined bit
Read: undefined value. Write: no effect.
TDRE Transmission data empty flag bit
0 The transmission data register (TDR) contains data.
1 The transmission data register is empty.
RDRF Reception data full flag bit
0 The reception data register (RDR) is empty.
1 The reception data register (RDR) contains data.
ORE Overrun error flag bit
0 No overrun error
1 Overrun error
Undefined bits
Read: undefined value. Write: no effect.
TSET Transmission buffer empty flag setting bit
Write Read
0 No effect "0" is always read.
1 Sets TDRE bit
REC Reception error flag clear bit
Write Read
0 No effect "0" is always read.
1 Clears the reception error flag (ORE)
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.19-6 Functional Description of Each Bit of Serial Status Register (SSR) (1 / 2)

Bit name		Function
bit15	REC: Reception error flag clear bit	This bit is used to clear the ORE bit in the serial status register (SSR). • Writing "1" clears the ORE bit. • Writing "0" has no effect. Reading this bit always returns "0".
bit14	TSET: Transmission buffer empty flag setting bit	This bit is used to set the TDRE bit in the serial status register (SSR). • Writing "1" sets the TDRE bit. • Writing "0" has no effect. Reading this bit always returns "0".

Table 28.19-6 Functional Description of Each Bit of Serial Status Register (SSR) (2 / 2)

Bit name		Function
bit13, bit12	Undefined bits	Read: undefined value Write: no effect
bit11	ORE: Overrun error flag bit	• This bit is set to "1" when an overrun occurs during reception. The bit is cleared by writing "1" to the REC bit in the serial status register (SSR). • A reception interrupt request is output when the ORE and RIE bits are set to "1". • When this flag is set, the data in the reception data register (RDR) is invalid. • If this flag is set during the use of the reception FIFO, the reception data will not be stored to the reception FIFO.
bit10	RDRF: Reception data full flag bit	• This flag indicates the status of the reception data register (RDR). • A reception interrupt request is output when the RIE bits and the reception data flag bit (RDRF) are set to "1". • The bit is set to "1" when reception data is loaded to RDR. The bit is cleared to "0" when the reception data register (RDR) is read. • This bit is set at the falling edge of SCL in the 8th bit of data. • It is also set by a NACK response. • RDRF is set to "1" when a specified number of data elements are received at the reception FIFO during the use of the reception FIFO. • This bit is cleared to "0" when the reception FIFO, if used, becomes empty. • During the use of the reception FIFO, RDRF will be set to "1", if the idle state of reception continues at the reception baud rate clock for a duration of eight clocks or longer as the specified number of data elements have not been received at the reception FIFO and some data still remains in the reception FIFO as well as the BER bit is set to "0". If RDR is read while 8 clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. Note: NACK response: indicates that SDA of the I^2C bus is at "H" during acknowledge.
bit9	TDRE: Transmission data empty flag bit	• This flag indicates the status of the transmission data register (TDR). • A transmission interrupt request is output when the TIE and TDRE bits are set to "1" • When transmission data is written to TDR, the bit becomes "0", indicating that TDR contains valid data. When the data is loaded to the transmission shift register and transmission starts, the bit becomes "1", indicating that TDR no longer contains any valid data. • This bit is set when "1" is written to the TSET bit in the serial status register (SSR). This bit is used to set the TDRE bit to "1" when an arbitration lost condition or bus error is detected.
bit8	Undefined bit	Read: undefined value Write: no effect

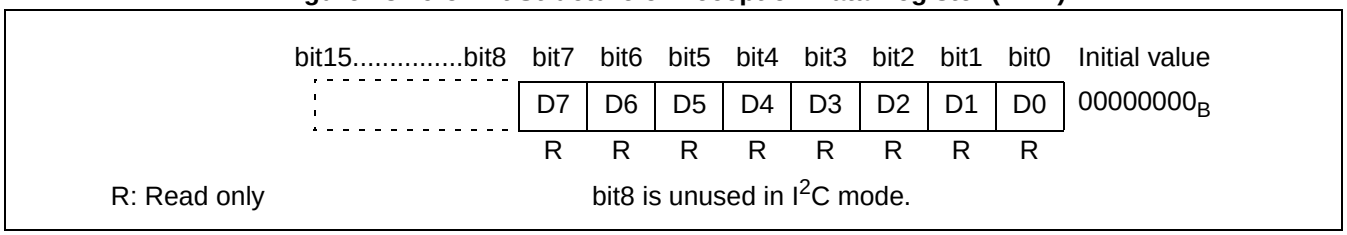
28.19.5 Reception Data Register / Transmission Data Register (RDR/TDR)

The reception data register and transmission data register are located at the same address. It serves as the reception data register in read access, while it functions as the transmission data register in write access.

■ Reception Data Register (RDR)

Figure 28.19-5 illustrates the bit structure of the serial reception register (RDR).

Figure 28.19-5 Bit Structure of Reception Data Register (RDR)



The reception data register (RDR) is a data buffer register for serial data reception.

- A serial data signal sent to a serial data line (SDA pin) is converted through the shift register and then stored in the reception data register (RDR).
- When the first byte* is received, the least significant bit (RDR: D0) becomes the data direction bit.
- The reception data full flag bit (SSR:RDRF) is set to "1" once reception data is stored in the reception data register (RDR).
- The reception data full flag bit (SSR: RDRF) is cleared to "0" automatically, when the reception data register (RDR) is read.

*: Indicates the data after a (repeated) start condition.

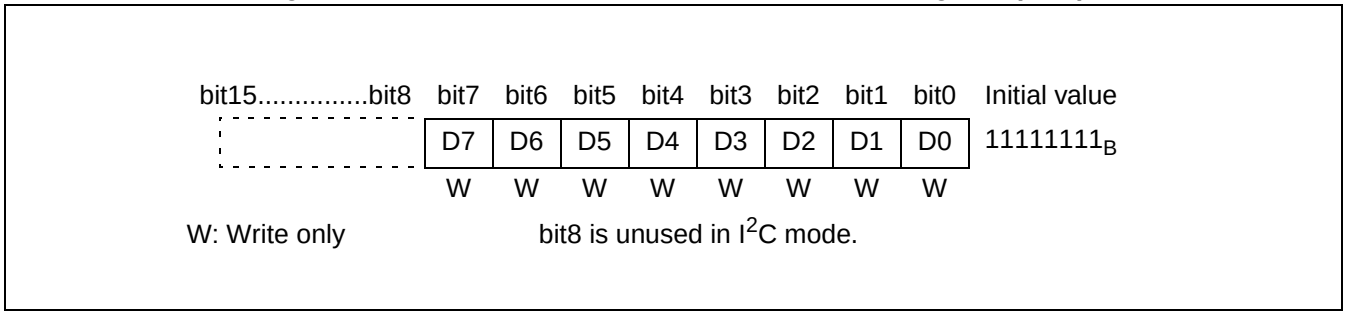
<Notes>

- RDRF is set to "1", once a specified number of data elements have been received at the reception FIFO, if used.
- RDRF is cleared to "0" when the reception FIFO, if used, becomes empty.

■ Transmission Data Register (TDR)

Figure 28.19-6 illustrates the bit structure of the transmission data register.

Figure 28.19-6 Bit Structure of Transmission Data Register (TDR)



The transmission data register (TDR) is a data buffer register for serial data transmission.

- Data is output to the serial data line (SDA pin), based on the transmission data register (TDR) value

(MSB first).

- The least significant bit (TDR: D0) becomes the data direction bit when transmitting the first byte.
- The transmission data empty flag (SSR: TDRE) is cleared to "0" when transmission data is written to the transmission data register (TDR).
- The transmission data empty flag (SSR: TDRE) is set to "1" when transmission data is transferred to the transmission shift register.
- Write the next transmission data under the following conditions.
  - The interrupt flag (INT bit) is set to "1".
  - No bus error is occurring (BER bit = 0).
  - Acknowledge is returned as ACK response ("0" is received as acknowledgement).
- Transmission data cannot be written to the transmission data register (TDR) if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is disabled.
- Transmission data can be written up to the capacity of the transmission FIFO, even if the data empty flag (SSR: TDRE) is set to "0" when the transmission FIFO is used.

<Note>

The transmission data register is used exclusively for writing, while the reception data register is used exclusively for reading. The two registers have different write and read values as they are located at the same address. Therefore, instructions such as INC/DEC instructions, which are used for read modify write (RMW) instruction, cannot be used.

28.19.6 7-bit Slave Address Mask Register (ISMK)

The 7-bit slave address mask register (ISMK) determines whether each bit of a slave address should be compared.

■ 7-bit Slave Address Mask Register (ISMK)

Figure 28.19-7 shows the bit structure of the 7-bit slave address mask register (ISMK) and Table 28.19-7 describes the function of each bit.

Figure 28.19-7 Bit Structure of 7-bit Slave Address Mask Register (ISMK)

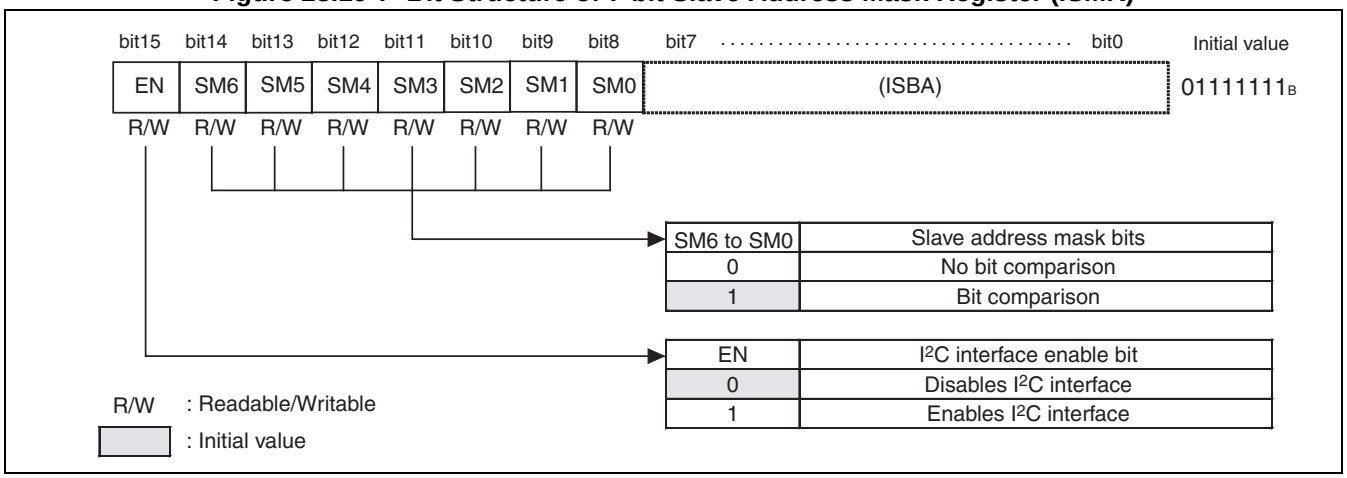

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
EN SM6 SM5 SM4 SM3 SM2 SM1 SM0 (ISBA) 01111111B
R/W R/W R/W R/W R/W R/W R/W R/W
SM6 to SM0 Slave address mask bits
0 No bit comparison
1 Bit comparison
EN I2C interface enable bit
0 Disables I2C interface
1 Enables I2C interface
R/W : Readable/Writable
: Initial value

Table 28.19-7 Functional Description of Each Bit of 7-bit Slave Address Mask Register (ISMK)

Bit name		Function
bit15	EN: I^2C interface enable bit	This bit is used to enable or disable the operation of the I^2C interface. Setting the bit to "0" disables the operation of the I^2C interface. Setting the bit to "1" enables the operation of the I^2C interface. Note: This bit is not cleared to "0" even when the BER bit in the IBSR register is set to "1". Set the baud rate generator when this bit is set to "0". Set a 7-bit slave address and the 7-bit slave mask register when this bit is set to "0". Setting the EN bit to "0" during transmission may generate a pulse at SDA/SCL of the I^2C bus. If FIFO has been enabled, write "0" to the EN bit after disabling FIFO.
bit14 to bit8	SM6 to SM0: Slave address mask bits	These bits are used to determine whether to compare the 7-bit slave address with the received address. Bit set to "1": compared Bit set to "0": handled as matched Note: Set this register when the EN bit is "0".

28.19.7 7-bit Slave Address Register (ISBA)

The 7-bit slave address register (ISBA) sets a slave address.

■ 7-bit Slave Address Register (ISBA)

Figure 28.19-8 shows the bit structure of the 7-bit slave address register (ISBA) and Table 28.19-8 describes the function of each bit.

Figure 28.19-8 Bit Structure of 7-bit Slave Address Register (ISBA)

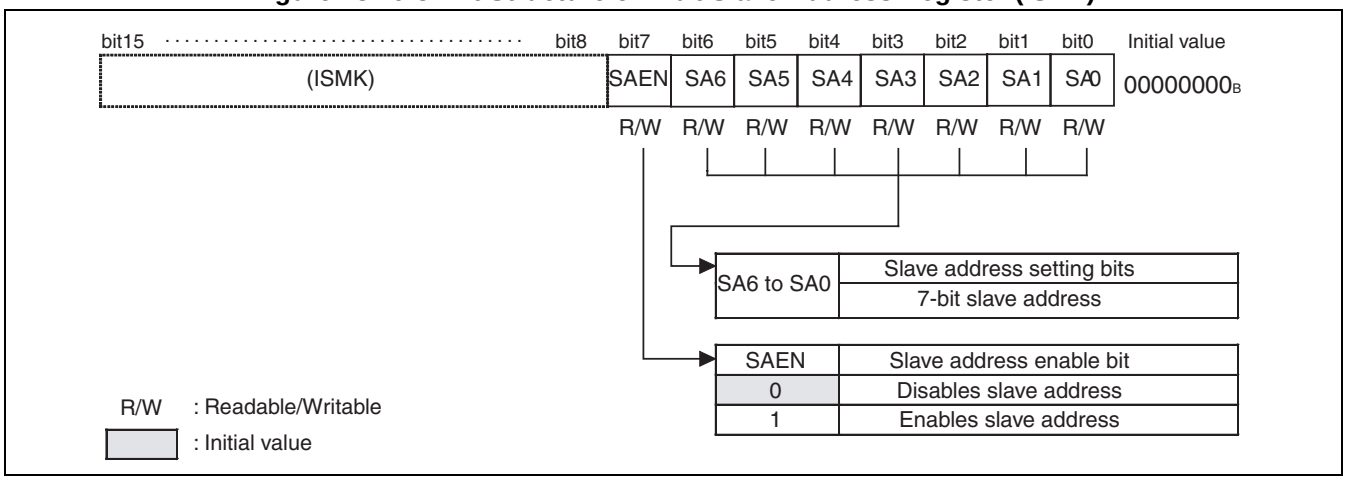


Table 28.19-8 Functional Description of Each Bit of 7-bit Slave Address Register (ISBA)

Bit name		Function
bit7	SAEN: Slave address enable bit	This bit is used to enable the detection of a slave address. Setting the bit to "0": Slave address not detected Setting the bit to "1": ISBA/ISMK setting compared with the first byte of received data
bit6 to bit0	SA6 to SA0: 7-bit slave address	If slave address detection has been enabled (SAEN = 1), the 7-bit data which is received after the detection of a (repeated) start condition will be compared with the value contained in the 7-bit slave address register (ISBA). If all the bits match, the device will operate in slave mode and output an ACK. At this point, the received slave address will be set to this register (An ACK will not be output if SAEN is set to "0"). The address bits which are set to "0" in the ISMK register are not subject to this comparison. Note: It is prohibited to set a reserved address. Set this register when the EN bit in the ISMK register is "0".

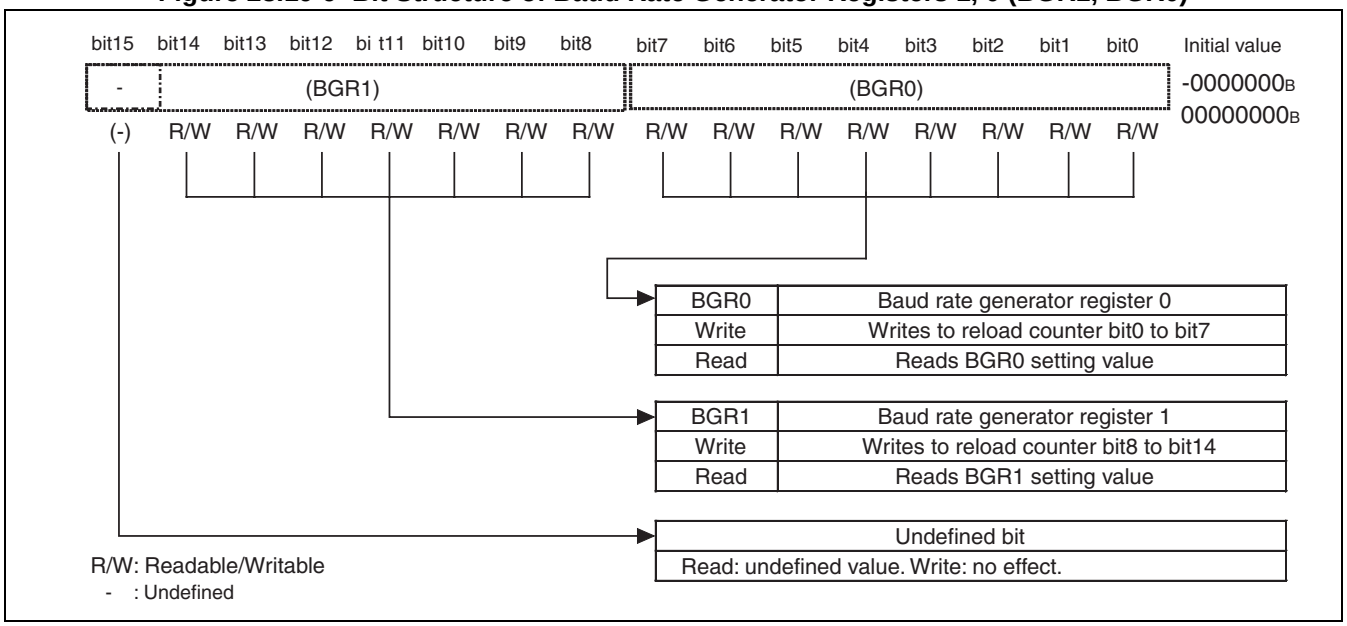
28.19.8 Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

The baud rate generator registers 1, 0 (BGR1, BGR0) are used to set a division ratio for the serial clock.

■ Bit Structure of the Baud Rate Generator Registers 1, 0 (BGR1, BGR0)

Figure 28.19-9 shows the bit structure of the baud rate generator registers 1, 0 (BGR1, BGR0).

Figure 28.19-9 Bit Structure of Baud Rate Generator Registers 1, 0 (BGR1, BGR0)


bit15 bit14 bit13 bit12 bi t11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
- (BGR1) (BGR0) -0000000B 00000000B
(-) R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
BGR0 Baud rate generator register 0
Write Writes to reload counter bit0 to bit7
Read Reads BGR0 setting value
BGR1 Baud rate generator register 1
Write Writes to reload counter bit8 to bit14
Read Reads BGR1 setting value
Undefined bit
Read: undefined value. Write: no effect.
R/W: Readable/Writable
- : Undefined

The baud rate generator registers are used to set a division ratio for the serial clock.

BGR0 and BGR1 correspond to the upper bits and lower bits respectively and they can write a reload value to be counted as well as read BGR1/BGR0 setting values.

The reload counter starts counting when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

<Notes>

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- Set the baud rate generator registers when the EN bit in the ISMK register is "0".
- Set a baud rate regardless of master or slave mode.
- Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

28.19.9 FIFO Control Register 1 (FCR1)

The FIFO control register 1 (FCR1) selects transmission/reception FIFO, enables transmission FIFO interrupts, and controls the interrupt flag.

■ Bit Structure of FIFO Control Register 1 (FCR1)

Figure 28.19-10 shows the bit structure of the FIFO control register 1 (FCR1) and Table 28.19-9 describes the function of each bit.

Figure 28.19-10 Bit Structure of FIFO Control Register 1 (FCR1)

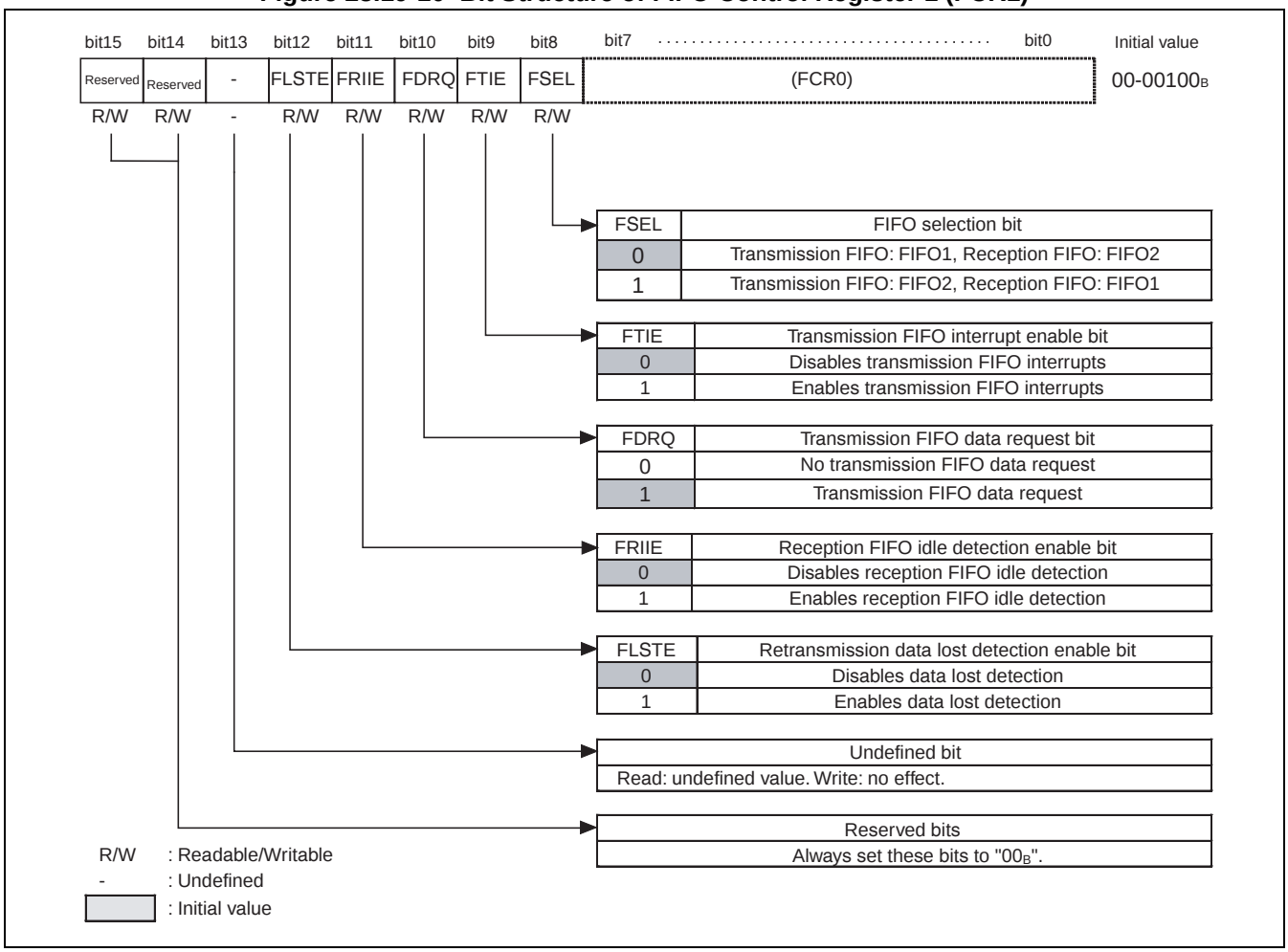

bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 ..... bit0 Initial value
Reserved Reserved - FLSTE FRIIE FDRQ FTIE FSEL (FCR0) 00-00100B
R/W R/W - R/W R/W R/W R/W R/W
FSEL FIFO selection bit
0 Transmission FIFO: FIFO1, Reception FIFO: FIFO2
1 Transmission FIFO: FIFO2, Reception FIFO: FIFO1
FTIE Transmission FIFO interrupt enable bit
0 Disables transmission FIFO interrupts
1 Enables transmission FIFO interrupts
FDRQ Transmission FIFO data request bit
0 No transmission FIFO data request
1 Transmission FIFO data request
FRIIE Reception FIFO idle detection enable bit
0 Disables reception FIFO idle detection
1 Enables reception FIFO idle detection
FLSTE Retransmission data lost detection enable bit
0 Disables data lost detection
1 Enables data lost detection
Undefined bit
Read: undefined value. Write: no effect.
Reserved bits
Always set these bits to "00B".
R/W : Readable/Writable
- : Undefined
: Initial value

Table 28.19-9 Functional Description of Each Bit of FIFO Control Register 1 (FCR1) (1 / 2)

Bit name		Function
bit15, bit14	Reserved bits	Always set these bits to "00_B".
bit13	Undefined bit	Read: undefined value Write: no effect

Table 28.19-9 Functional Description of Each Bit of FIFO Control Register 1 (FCR1) (2 / 2)

Bit name		Function
bit12	FLSTE: Retransmission data lost detection enable bit	This bit enables FLST bit detection. Setting the bit to "0" disables FLST bit detection. Setting the bit to "1" enables FLST bit detection. Note: To set this bit to "1", set the FSET bit to "1" beforehand.
bit11	FRIIE: Reception FIFO idle detection enable bit	This bit is used to determine whether the idle state of reception for 8 clocks with the baud rate clock or longer should be detected while the reception FIFO still contains valid data. A reception interrupt will occur if the idle state of reception is detected when reception interrupts have been enabled (SCR:RIE = 1). Setting the bit to "0" disables reception idle state detection. Setting the bit to "1" enables reception idle state detection.
bit10	FDRQ: Transmission FIFO data request bit	This is a transmission FIFO data request bit. When this bit is set to "1", it is indicated that transmission data is being requested. If transmission interrupts have been enabled (FTIE = 1) at this point, a FIFO transmission interrupt request will be output. FDRQ setting condition • FBYTE1/FBYTE2 (for transmission) = 0 (The transmission FIFO is empty.) • Transmission FIFO reset FDRQ reset condition • Writing "0" to this bit • When the transmission FIFO is full. Note: It is prohibited to write "0" to this bit when FBYTE1/FBYTE2 (for transmission) is set to "0". It is prohibited to modify the FSEL bit when this bit is set to "0". Writing "1" to the bit has no effect on operation. "1" is read by a read modify write (RMW) instruction.
bit9	FTIE: Transmission FIFO interrupt enable bit	This is a transmission FIFO interrupt enable bit. An interrupt will occur if this bit is set to "1" when the FDRQ bit is set to "1".
bit8	FSEL: FIFO selection bit	This bit is used to select transmission/reception FIFO. Setting the bit to "0" assigns transmission FIFO to FIFO1 and reception FIFO to FIFO2. Setting the bit to "1" assigns transmission FIFO to FIFO2 and reception FIFO to FIFO1. Note: This bit cannot be cleared by resetting FIFO (FCL2, FCL1 = 1). To modify this bit, disable FIFO operation (FCR0:FE2, FE1 = 0) in advance.

28.19.10 FIFO Control Register 0 (FCR0)

The FIFO control register 0 (FCR0) enables/disables FIFO operation, resets FIFO, saves the read pointer and sets retransmission.

■ Bit Structure of FIFO Control Register 0 (FCR0)

Figure 28.19-11 shows the bit structure of the FIFO control register 0 (FCR0) and Table 28.19-10 describes the function of each bit.

Figure 28.19-11 Bit Structure of FIFO Control Register 0 (FCR0)

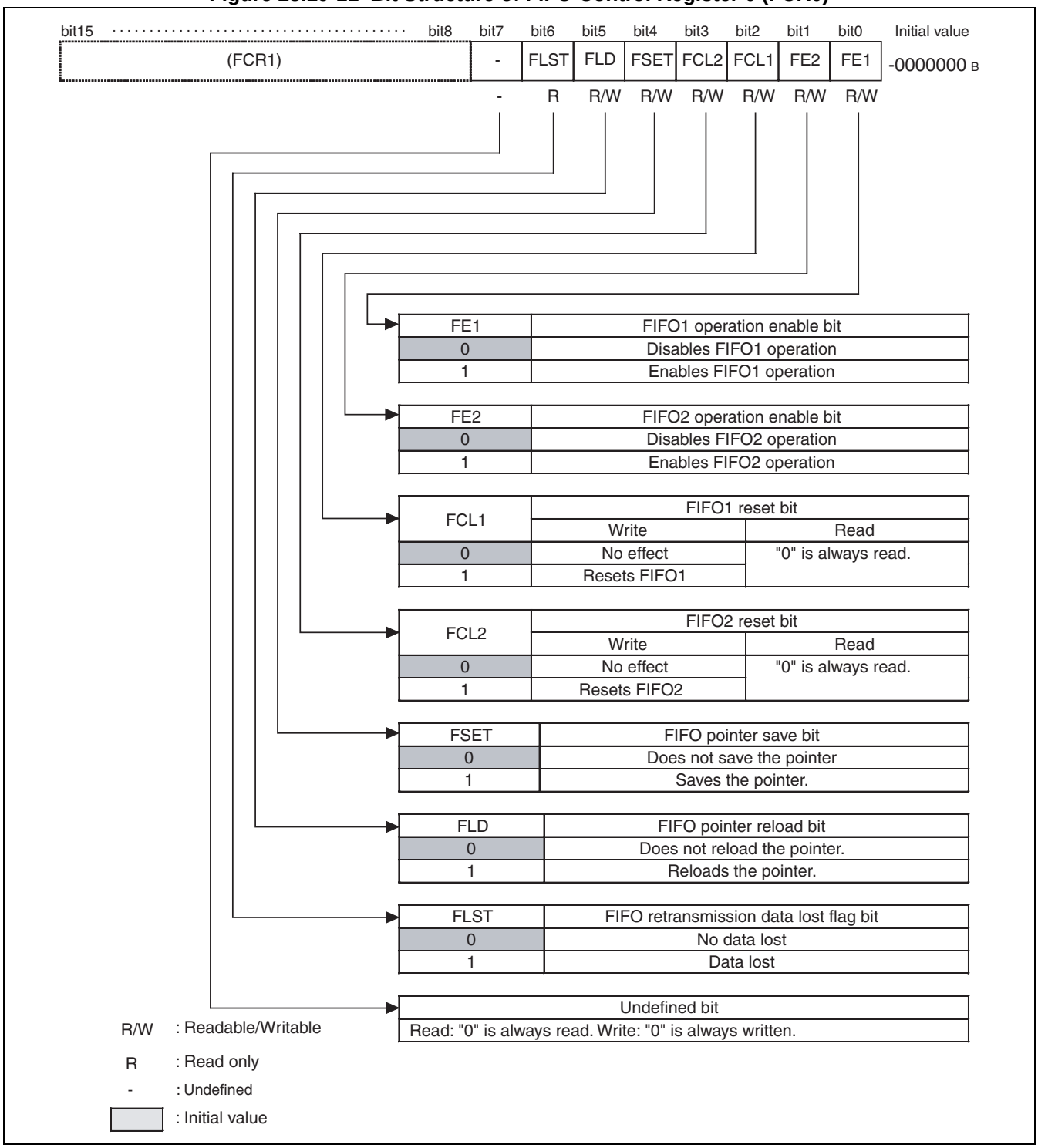

bit15 ..... bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FCR1) - FLST FLD FSET FCL2 FCL1 FE2 FE1 -0000000 B
- R R/W R/W R/W R/W R/W R/W
FE1 FIFO1 operation enable bit
0 Disables FIFO1 operation
1 Enables FIFO1 operation
FE2 FIFO2 operation enable bit
0 Disables FIFO2 operation
1 Enables FIFO2 operation
FCL1 FIFO1 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO1
FCL2 FIFO2 reset bit
Write Read
0 No effect "0" is always read.
1 Resets FIFO2
FSET FIFO pointer save bit
0 Does not save the pointer
1 Saves the pointer.
FLD FIFO pointer reload bit
0 Does not reload the pointer.
1 Reloads the pointer.
FLST FIFO retransmission data lost flag bit
0 No data lost
1 Data lost
Undefined bit
Read: "0" is always read. Write: "0" is always written.
R/W : Readable/Writable
R : Read only
- : Undefined
: Initial value

Table 28.19-10 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (1 / 3)

Bit name		Function
bit7	Undefined bit	Read: "0" is always read. Write: Always write "0".
bit6	FLST: FIFO retransmission data lost flag bit	This bit indicates that retransmission data has been lost from the transmission FIFO. FLST setting condition Writing to FIFO when the FLSTE bit in the FIFO control register 1 (FCR1) is set to "1" and also the write pointer of the transmission FIFO matches the read pointer saved by the FSET bit. FLST reset conditions • FIFO reset (writing "1" to FCL) • Writing "1" to the FSET bit Setting this bit to "1" overwrites the data indicated by the read pointer which has been saved by the FSET bit. Consequently, the FLD bit cannot be used to set retransmission even when an error occurs. To resend the data while this bit is set to "1", reset FIFO and then rewrite the data to FIFO.
bit5	FLD: FIFO pointer reload bit	This bit is used to reload to the read pointer the data which has been saved by the FSET bit to the transmission FIFO. This bit should be used to resend data when a communication error occurs. The bit becomes "0" when retransmission has been set. Note: Reload to the read pointer is in progress as long as this bit is set to "1". Therefore, do not perform write operations except for FIFO reset. It is prohibited to set this bit to "1" when FIFO has been enabled or transmission is in progress. Write "1" to this bit after setting the TIE bit to "0". And then, set the TIE bit to "1" when the transmission FIFO has been enabled.
bit4	FSET: FIFO pointer save bit	This bit is used to save the read pointer of the transmission FIFO. If the FLST bit is set to "0", saving the read pointer prior to transmission will enable retransmission in case that an error such as a communication error occurs. Setting the bit to "1" saves the current read pointer value. Setting the bit to "0" has no effect. Note: Set this bit to "1" when the number of transmission bytes (FBYTE1/FBYTE2) indicates "0".
bit3	FCL2: FIFO2 reset bit	This bit is used to reset FIFO2. Setting this bit to "1" initializes the internal state of FIFO2. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable FIFO2 before resetting it. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE2 register will become "0".

Table 28.19-10 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (2 / 3)

Bit name		Function
bit2	FCL1: FIFO1 reset bit	This bit is used to reset FIFO1. Setting this bit to "1" initializes the internal state of FIFO1. Only the FCR0:FLST bit will be initialized and the other bits in the FCR1/FCR0 registers will be retained. Note: Disable FIFO1 before resetting it. Set the transmission FIFO interrupt enable bit to "0" before the reset. The number of valid data elements for the FBYTE1 register will become "0".
bit1	FE2: FIFO2 operation enable bit	This bit is used to enable/disable FIFO2 operation. • To use FIFO2, set this bit to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO2 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO2 is disabled, its status is retained. Note: Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKE to "0". It the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO2 contains data.

Table 28.19-10 Functional Description of Each Bit of FIFO Control Register 0 (FCR0) (3 / 3)

Bit name		Function
bit0	FE1: FIFO1 operation enable bit	This bit is used to enable/disable FIFO1 operation. • To use FIFO1, set this bit to "1". • This bit will be cleared to "0" if a reception error occurs when FIFO1 has been selected as the reception FIFO by the FSEL bit. This bit cannot be set to "1" unless the reception error is cleared. • Set this bit to "1" or "0" when the transmission data is empty (TDRE = 1) to use it as the transmission FIFO, or when the reception data is empty (RDRF = 0) to use it as the reception FIFO. • Even when FIFO1 is disabled, its status is retained. Note: Switch between the enabling and disabling when the BB bit is "0" or the INT bit is "1". To allow the device to operate in slave transmission when it has been selected as the reception FIFO to detect a reserved address, use an interrupt generated by the detection of a reserved address to set this bit to "0" and ACKE to "0". It the RDRF bit in SSR is set to "1" when the device is used as the reception FIFO and this bit is switched from "1" to "0", the reception FIFO will not be disabled until the RDRF bit is set to "0". To switch this bit from "0" to "1", set the TIE bit to "0" first, write "1" to this bit and then set the TIE bit to "1", when the device is used as the transmission FIFO and FIFO1 contains data.

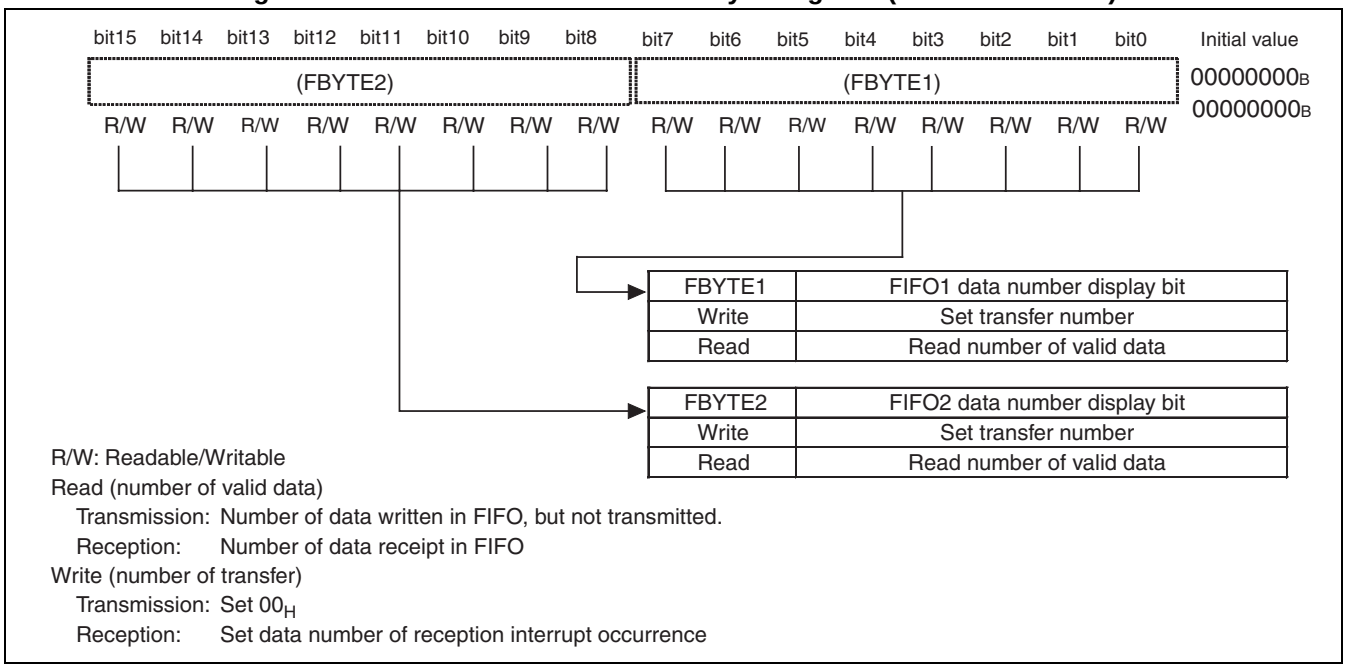
28.19.11 FIFO Byte Register (FBYTE1/FBYTE2)

The FIFO byte register (FBYTE1/FBYTE2) indicates the number of valid data elements for FIFO. This register can also be used to determine whether a reception interrupt should occur when the reception FIFO receives a specified number of data elements.

■ Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)

Figure 28.19-12 shows the bit structure of FIFO byte register (FBYTE1/FBYTE2).

Figure 28.19-12 Bit Structure of FIFO Byte Register (FBYTE1/FBYTE2)


bit15 bit14 bit13 bit12 bit11 bit10 bit9 bit8 bit7 bit6 bit5 bit4 bit3 bit2 bit1 bit0 Initial value
(FBYTE2)
(FBYTE1)
00000000B
00000000B
R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W R/W
FBYTE1 FIFO1 data number display bit
Write Set transfer number
Read Read number of valid data
FBYTE2 FIFO2 data number display bit
Write Set transfer number
Read Read number of valid data
R/W: Readable/Writable
Read (number of valid data)
Transmission: Number of data written in FIFO, but not transmitted.
Reception: Number of data receipt in FIFO
Write (number of transfer)
Transmission: Set 00H
Reception: Set data number of reception interrupt occurrence

The FBYTE1/FBYTE2 register indicates the number of valid data elements for FIFO. The details are as follows, depending on the FCR1:FSEL bit setting.

Table 28.19-11 Displaying the Number of Data Elements

FSEL	FIFO selection	Displaying the number of data elements
0	FIFO2: Reception FIFO, FIFO1: Transmission FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1
1	FIFO2: Transmission FIFO, FIFO1: Reception FIFO	FIFO2: FBYTE2, FIFO1: FBYTE1

- The initial value for the number of transfers at the FBYTE1/FBYTE2 register is "08_H".
- The number of data elements that will generate a reception interrupt flag in FBYTE1/FBYTE2 of the reception FIFO should be selected. When the selected number of transfers matches the displayed number of data elements in the FBYTE1/FBYTE2 register, the interrupt flag (RDRF) is set to "1".

- When the reception FIFO idle detection enable bit (FRIIE) is set to "1" and the number of data elements contained in the reception FIFO does not reach the number of transfers, the interrupt flag (RDRF) will be set to "1", if the idle state of reception continues at the baud rate clock for a duration of eight clocks or longer. If RDR is read while the eight clocks are still being counted, the counter will be reset to "0" and it will start counting another set of eight clocks. The counter will be reset to "0" if the reception FIFO is disabled. The count will restart if the reception FIFO is enabled when it still contains some data.
- To receive data in master operation (master reception), set the TIE bit to "0", set the number of data elements to be received to the FBYTE1/FBYTE2 register of the transmission FIFO, and write "0" to the FDRQ bit. The SCL clock will be output for a specified amount of data, and then the INT bit will be set to "1". To set the TIE bit to "1", wait until the FDRQ becomes "1".

<Notes>

- In master operation, set "00_H" to FBYTE1/FBYTE2 of the transmission FIFO except when receiving data.
- Set the number of transmission data elements when receiving data in master operation, when the transmission FIFO is empty and the TIE bit is set to "0".
- To disable the I^2C interface (EN = 0) during data reception in master operation, disable transmission/reception FIFO first.
- Select "1" or a larger data value for FBYTE1/FBYTE2 of the reception FIFO.
- Modify the register after disabling transmission/reception.
- Read modify write (RMW) instructions cannot be used for this register.
- Settings that will exceed the capacity of FIFO are prohibited.

28.20 Interrupts of I²C Interface

The following sources can be used to generate interrupt requests for the I²C interface.

- After the transmission/reception of the first byte or data
- Stop condition
- Repeated start condition
- FIFO transmission data request
- Completion of FIFO reception data

■ Interrupts of I²C Interface

Table 28.20-1 shows the interrupt control bits and interrupt sources of the I²C interface.

Table 28.20-1 Interrupt Control Bits and Interrupt Sources of I²C Interface (1 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Recep-tion	INT	IBCR	After transmission/reception of 1st byte*1	IBCR:INTE	Writing "0" to interrupt flag bit (IBCR:INT)
			After transmission/reception of data*1		
			Detection of bus error		
			Detection of arbitration lost condition		
			Reception of the amount set by FBYTE1/FBYTE2		Writing "0" to interrupt flag bit (IBCR:INT) after reading reception data (RDR) until reception FIFO becomes empty
			Detection of the idle state of reception for 8 clocks with the baud rate clock or longer while FRIIE bit is "1" and reception FIFO contains valid data		
	RDRF	SSR	Detection of reserved address	SMR:RIE	Reading reception data (RDR)
			After reception of data		
			Reception of the amount set by FBYTE1/FBYTE2		Reading reception data (RDR) until reception FIFO becomes empty
	ORE	SSR	Overrun error		Writing "1" to reception error flag bit (SSR:REC)
	SPC	IBSR	Stop condition	IBCR:CNDE	Writing "0" to stop condition detection bit
	RSC	IBSR	Repeated start condition		Writing "0" to repeated start detection flag bit (IBSR:RSC)

Table 28.20-1 Interrupt Control Bits and Interrupt Sources of I^2C Interface (2 / 2)

Interrupt type	Interrupt request flag bit	Flag register	Interrupt source	Interrupt source enable bit	Clearing of interrupt request flag
Trans-mission	TDRE	SSR	Transmission register being empty	SMR:TIE	Writing to transmission data (TDR), or writing "1" to the transmission FIFO operation enable bit while the transmission FIFO operation enable bit is "0" and the transmission FIFO contains valid data (retransmission)*2
			Writing "1" to transmission buffer empty flag setting bit (SSR:TSET)		
	FDRQ	FCR1	Transmission FIFO being empty	FCR1:FTIE	Writing "0" to the FIFO transmission data request bit, or transmission FIFO being full

*1: Normal data can be transmitted or received. An interrupt does not occur when TDRE is set to "0". This function is designed to support DMA transfer.
The TDRE bit must be set to "1" before the INT flag is set, in order to generate the INT flag in data transmission/reception.

*2: Wait until the TDRE bit becomes "0" before setting the TIE bit to "1".

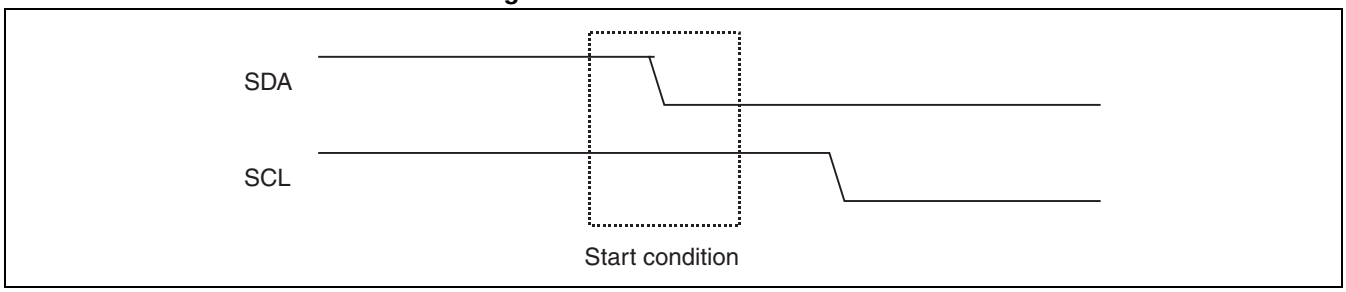
28.20.1 Operation of I²C Interface Communication

The I²C interface communication uses 2 two-way bus lines, a serial data line (SDA) and a serial clock line (SCL).

■ Start Condition for I²C Bus

The start condition for the I²C bus is shown below.

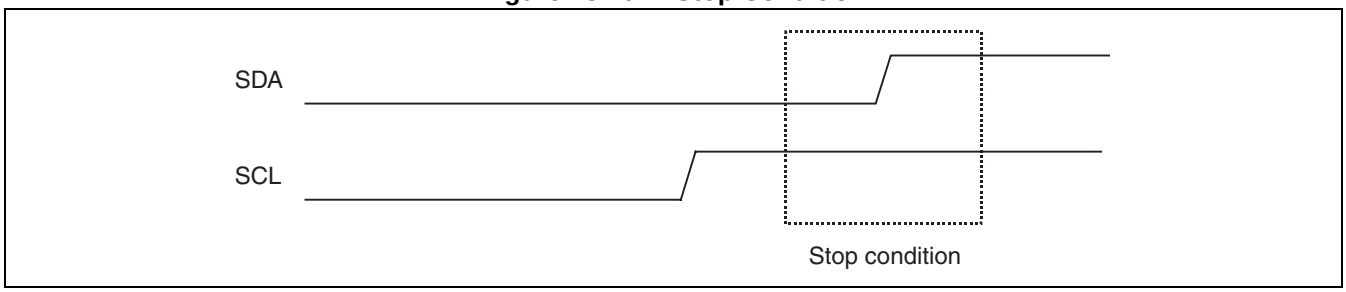
Figure 28.20-1 Start Condition



■ Stop Condition for I²C Bus

The stop condition for the I²C bus is shown below.

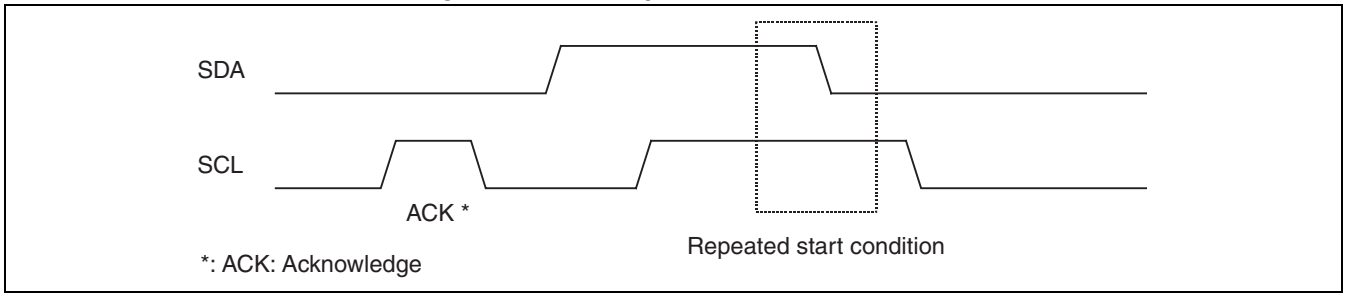
Figure 28.20-2 Stop Condition



■ Repeated Start Condition for I²C Bus

The repeated start condition for the I²C bus is shown below.

Figure 28.20-3 Repeated Start Condition



28.20.2 Master Mode

Master mode generates a start condition for the I^2C bus and outputs a clock to the I^2C bus. Master mode will be selected and the ACT bit in the IBCR register will be set to "1", if the MSS bit in the IBCR register is set to "1" when the I^2C bus is in an idle state (SCL = "H", SDA = "H").

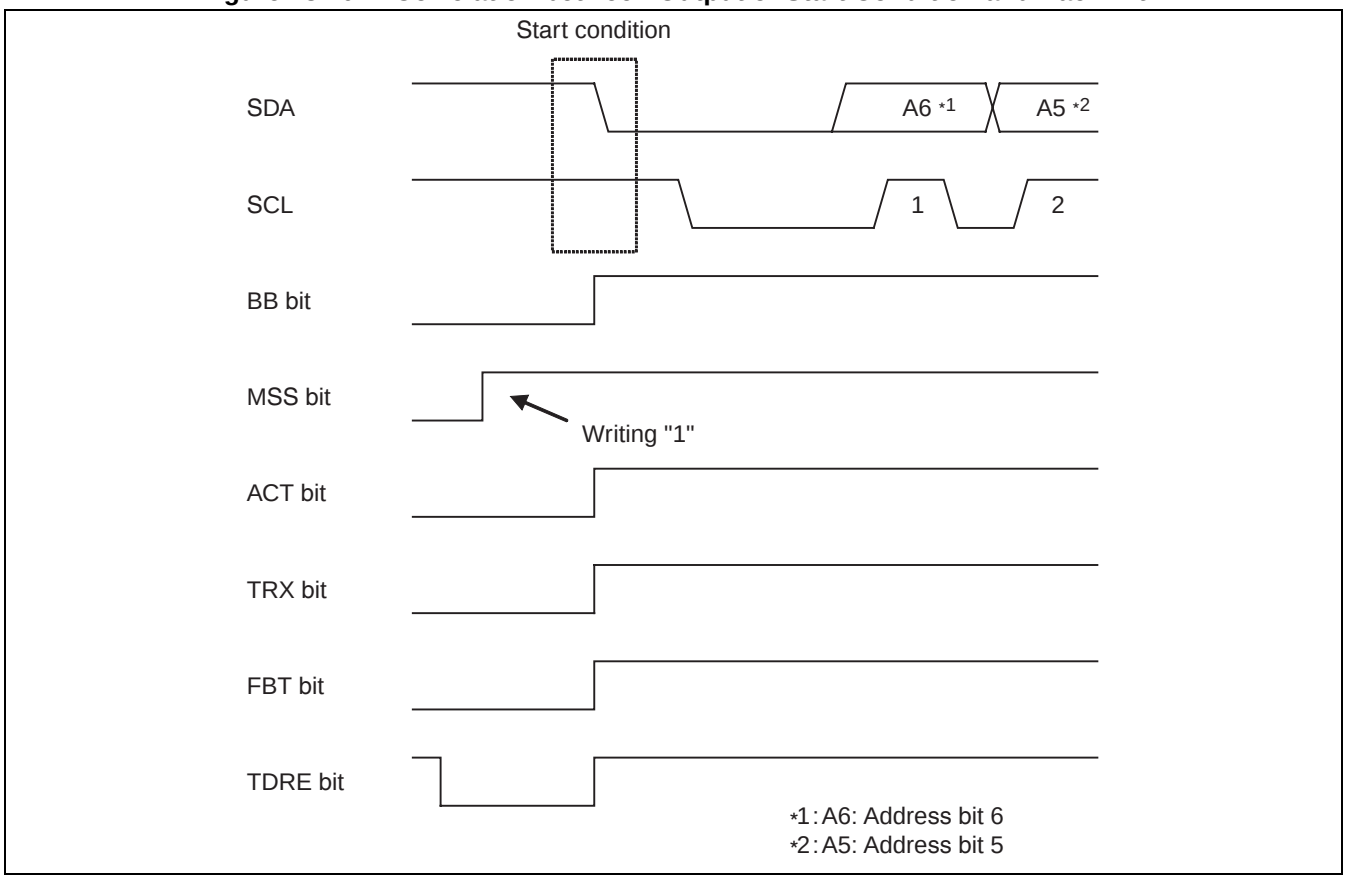
■ Generating a Start Condition

A start condition is output under the following conditions.

"1" is written to the MSS bit when SDA = "H", SCL = "H", EN = 1, and BB = 0.

Outputting a start condition to the I^2C bus sets the ACT bit to "1". After that, the BB bit is set to "1", indicating that the I^2C bus is in the middle of communication, once the start condition is received (see Figure 28.20-4).

Figure 28.20-4 Correlation between Output of Start Condition and Each Bit



<Note>

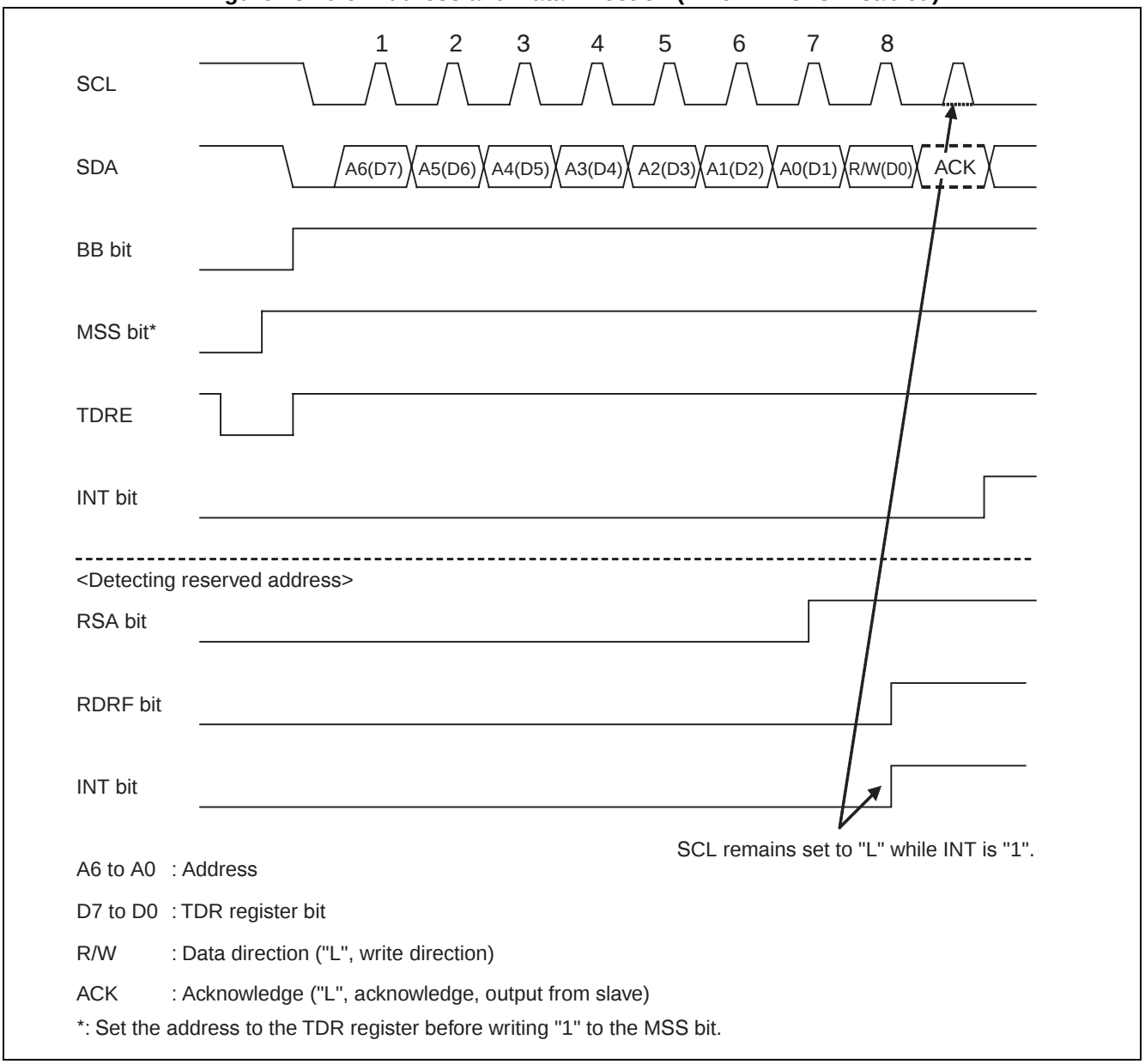
Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.

■ Outputting a Slave Address

When a start condition is output, the data set in the TDR register is output from bit7 as an address. When FIFO has been enabled, the first data written in the TDR register is output. Bit0 is used as the data direction bit (R/W), and the data indicates the write direction (master → slave) when the data direction bit (R/W) is set to "0". Set an address to the TDR register before writing "1" to MSS or SCC.

Figure 28.20-5 and Figure 28.20-6 show the address and data direction output timings.

Figure 28.20-5 Address and Data Direction (When FIFO is Disabled)



A6 to A0 : Address

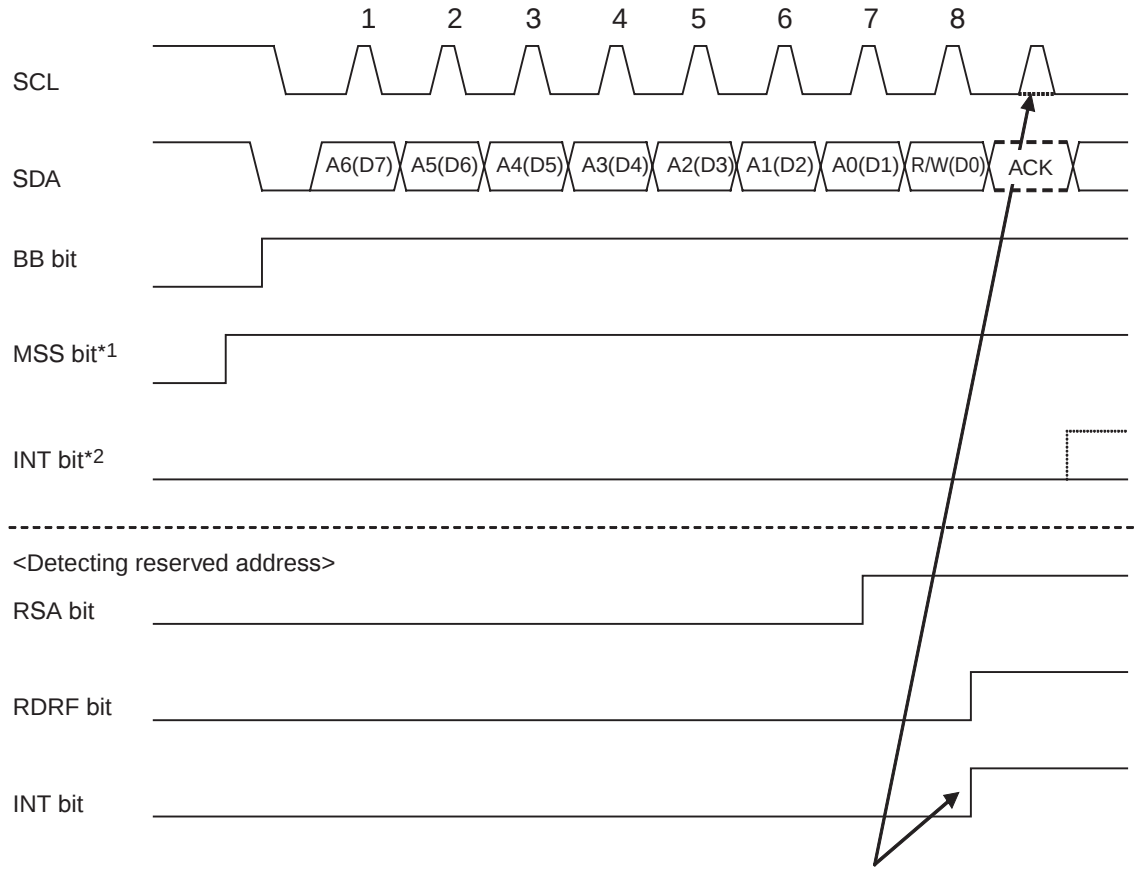
D7 to D0 : TDR register bit

R/W : Data direction ("L", write direction)

ACK : Acknowledge ("L", acknowledge, output from slave)

*: Set the address to the TDR register before writing "1" to the MSS bit.

Figure 28.20-6 Address and Data Direction (When Transmission/Reception FIFO is Enabled)



A6 to A0 : Address

D7 to D0 : TDR register bit

R/W : Data direction ("L", write direction)

ACK : Acknowledge ("L", acknowledge, output from slave)

*1: Set the address to the TDR register before writing "1" to the MSS bit.
*2: The INT bit will not be set to "1" if the transmission FIFO contains data when acknowledge is "L" and R/W is "L" or if the reception FIFO contains no data when acknowledge is "L" and R/W is "H".

■ Receiving Acknowledge after Transmitting 1st Byte

The I^2C interface receives an acknowledge from the slave when the data direction bit (R/W) is output. The following operations are performed when FIFO is enabled and disabled.

Table 28.20-2 Operations after Reception of Acknowledge (RSA Bit = 0)

Trans-mission FIFO	Reception FIFO	Trans-mission FIFO status	Receptio n FIFO status	Data direction bit (R/W)	Operation immediately after reception of acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
				1		
Disabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data con-tained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	
Enabled	Disabled	-	-	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
				1		
Enabled	Enabled	-	No data contained	0	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	The INT bit is set to "1", causing a wait
			Data con-tained		The INT bit is set to "1", causing a wait	
			-	1	When the TDRE bit is set to "1", the INT bit is set to "1", causing a wait. When the TDRE bit is set to "0", the INT bit remains "0", causing no wait	

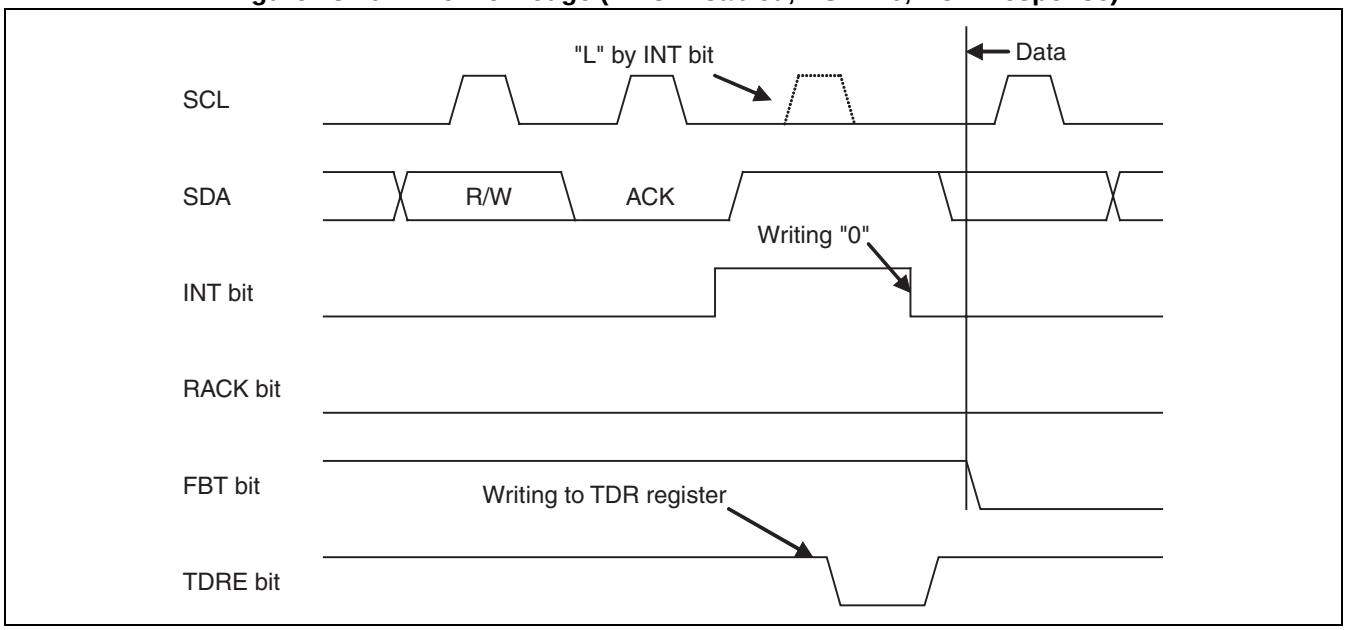
● FIFO disabled (both transmission FIFO and reception FIFO disabled)

- The interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L", if the TDRE bit is set to "1" after the reception of an acknowledge when the RSA bit is set to "0". The wait is cancelled when "0" is written to the interrupt flag to set it to "0". If the TDRE bit has been set to "0", a clock will be generated to SCL without setting the interrupt flag to "1" when an ACK is received.
- When the RSA bit is set to "1", the interrupt flag (INT) is set to "1", causing a wait while maintaining SCL at "L", after a reserved address is received (before acknowledge). The interrupt flag will be set to "0" to cancel the wait, if the ACKE bit and transmission data are set and "0" is written to the interrupt flag after the RDR register has been read.
- The received acknowledge is set to the RACK bit. If NACK is identified when the RACK bit is checked during the wait, a stop condition or repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

● FIFO enabled

- The following FIFO settings must be performed before the MSS bit is set to "1".
  - For transmission to the slave (data direction bit = 0), data including a slave address should be set to the transmission FIFO.
  - For reception of data from the slave (data direction bit = 1), the number of receptions should be set to the FIFO byte number register to write to the transmission data register using dummy data for the number of data elements to be received for the slave address and data direction bit.
- When the RSA bit is set to "0", the interrupt flag (INT) will not be set to "1" and data will be transmitted/received according to the data direction bit, if the received acknowledge is an ACK (no wait). If the acknowledge is a NACK, the interrupt flag (INT) will be set to "1", causing a wait while maintaining SCL at "L".
- The received acknowledge is stored in the RACK bit. If the acknowledge is a NACK when the RACK bit is checked during the wait, a stop condition or a repeated start condition will be generated by writing "0" to the MSS bit or writing "1" to the SCC bit. In this case, the INT bit will be cleared to "0" automatically.

Figure 28.20-7 Acknowledge (FIFO Disabled, RSA = 0, ACK Response)



Address wait timings:

- RSA = 0: after receiving acknowledge
- RSA = 1: before receiving acknowledge

The above timings are not dependent on the WSEL setting.

Figure 28.20-8 Acknowledge (FIFO Disabled, RSA = 0, NACK Response)

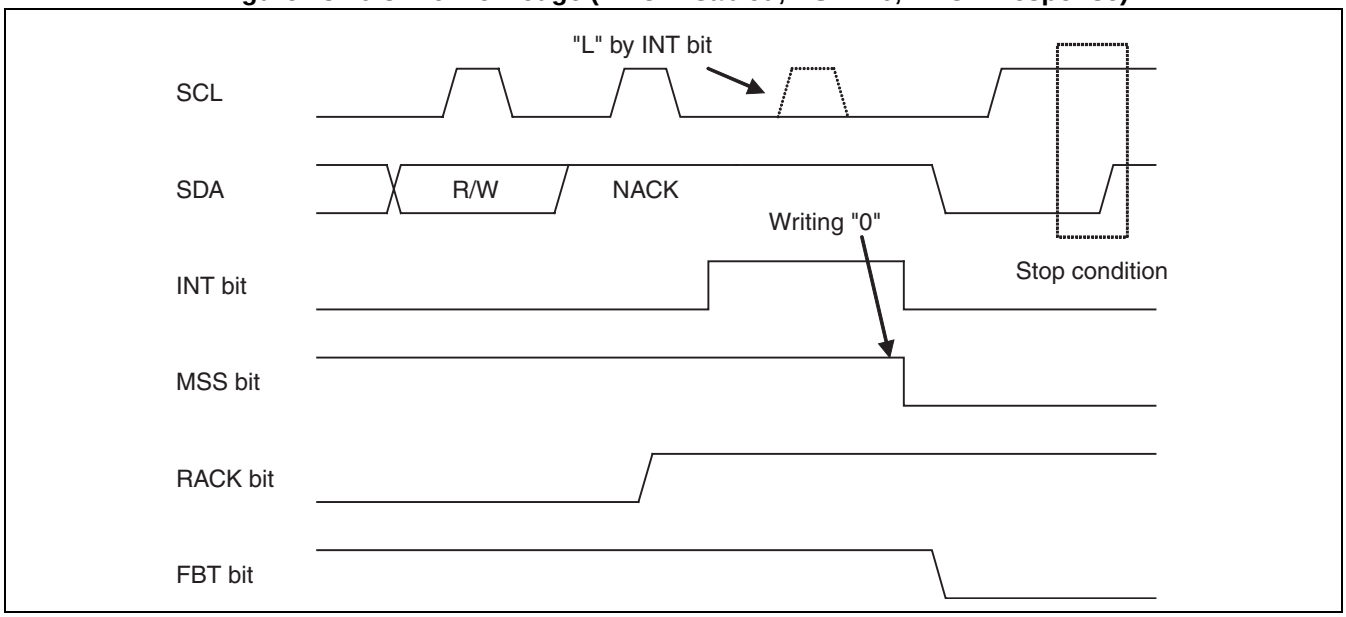


Figure 28.20-9 Acknowledge (FIFO Disabled, RSA = 1, ACK Response)

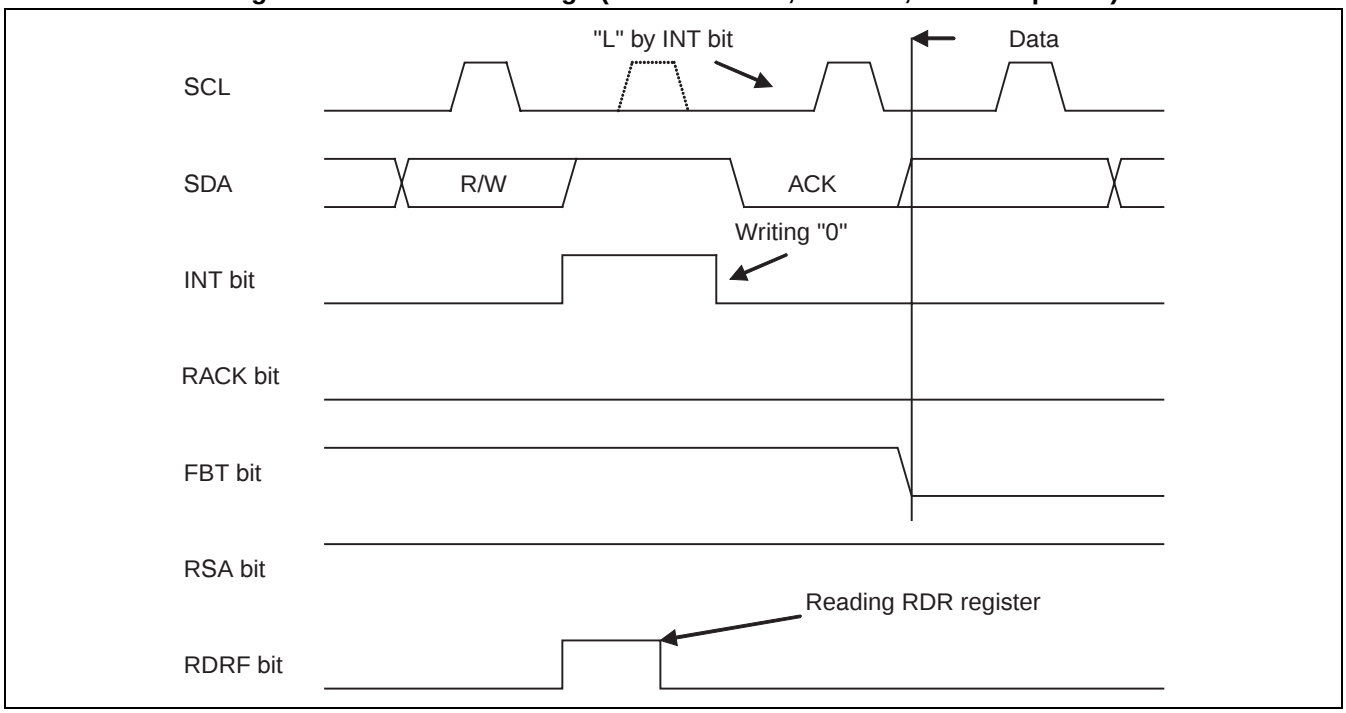


Figure 28.20-10 Acknowledge (FIFO Disabled, RSA = 1, NACK Response)

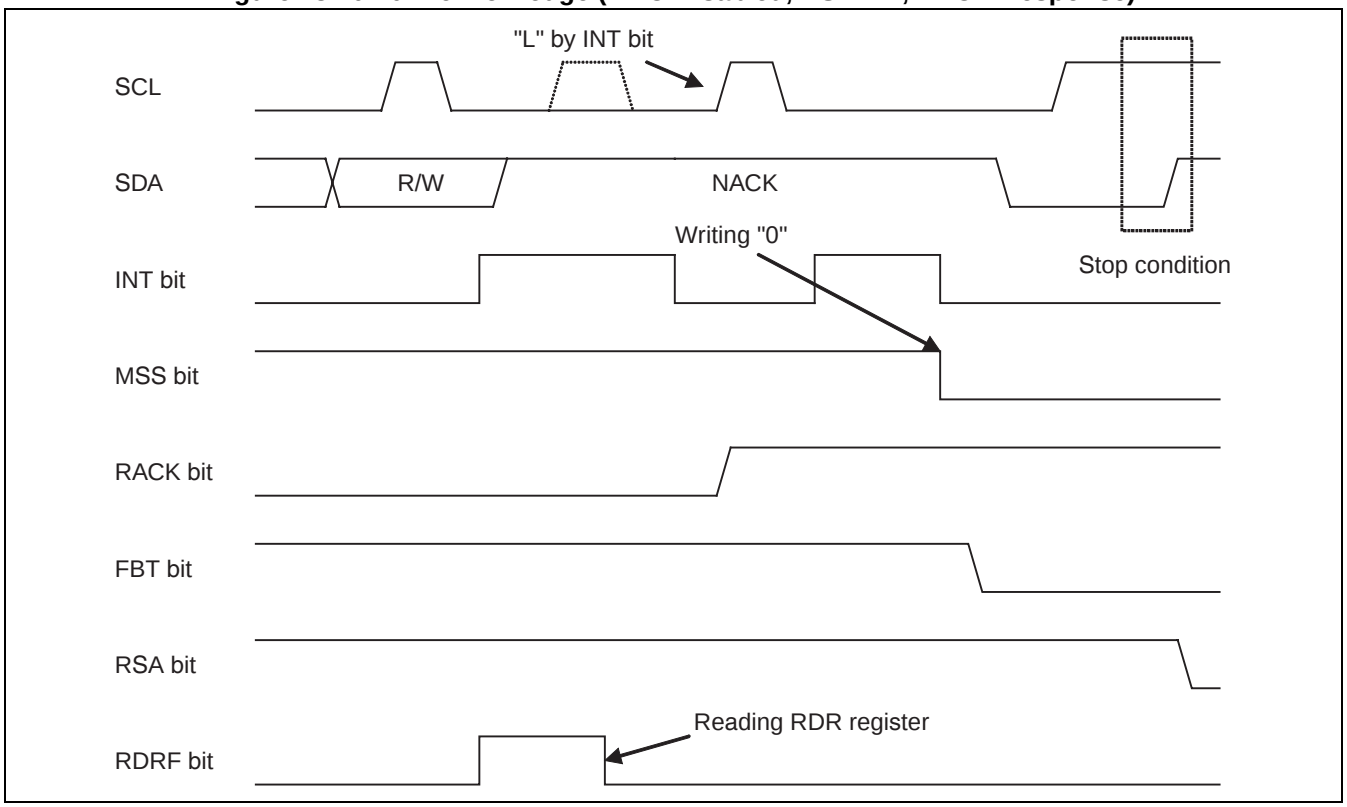
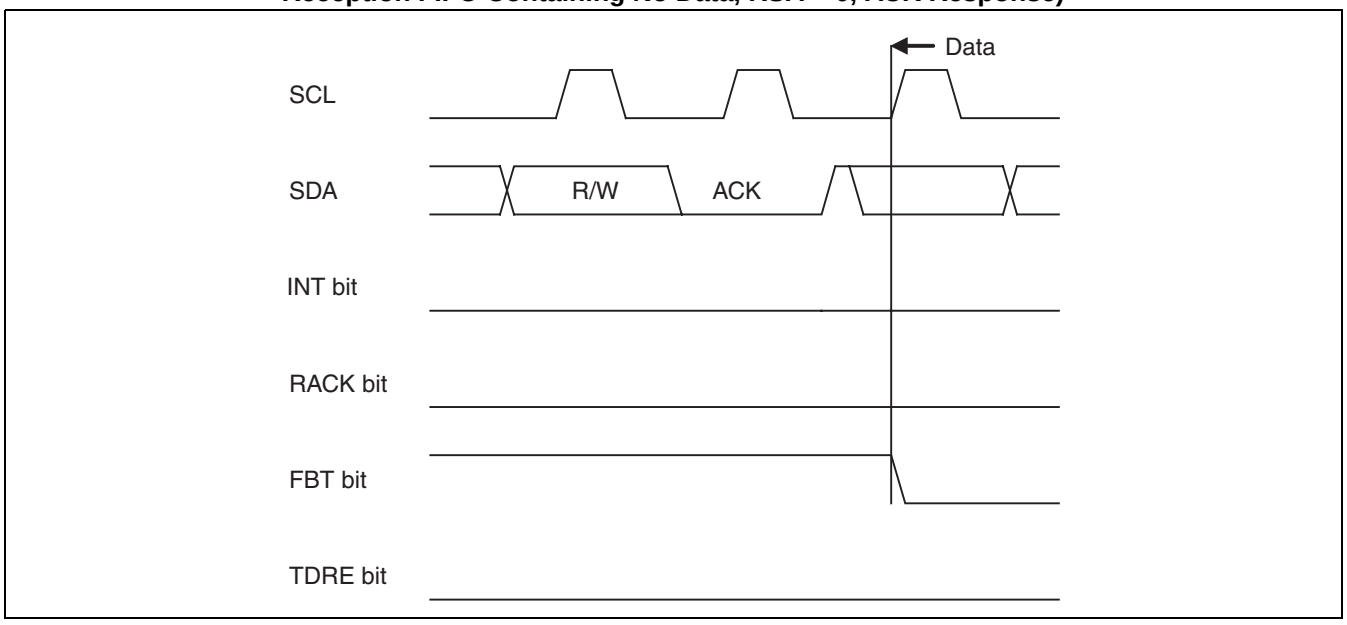


Figure 28.20-11 Acknowledge (FIFO Enabled, Transmission FIFO Containing Data, Reception FIFO Containing No Data, RSA = 0, ACK Response)



■ Master Data Transmission

Data is transmitted from the master when the data direction bit (R/W) is set to "0". The slave returns an ACK or NACK response for each byte transmitted.

The location in which a wait occurs is as follows, depending on the WSEL bit setting.

Table 28.20-3 WSEL Bit During Master Data Transmission

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after an acknowledge by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1" or detecting an arbitration lost condition. If FIFO is enabled, a wait will be generated when the interrupt flag (INT) is set to "1" after data transmission by detecting an arbitration lost condition, or when the transmission data register no longer contains valid data (TDRE = 1).

However, if a NACK is received at times other than when a stop condition is set (MSS = 0, MAS = 1), the interrupt flag (INT) is set after an acknowledge, regardless of the WSEL setting.

An example procedure for transmitting data to the slave is shown below.

● Transmitting data to any address other than reserved address

- When transmission FIFO is disabled:
  (1) Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
  (2) An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
  (3) Write the data to be transmitted to the TDR register.
  (4) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I^2C bus.
  (5) Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".
  (6) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.
- When transmission FIFO is enabled:
  (1) Write the slave address (including the data direction bit) and transmission data to the TDR register.
  (2) Set the WSEL bit and write "1" to the MSS bit.
  (3) If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I^2C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I^2C bus in a wait.
  (4) Write "0" to the MSS bit to generate a stop condition.

● Transmitting data to reserved address

- When transmission FIFO is disabled:

  (1) Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.

  (2) The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.

  (3) Read from the RDR register to check the reserved address.*

  (4) Write the data to be transmitted to the TDR register.

  (5) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel the wait for the I^2C bus.

  (6) Put the I^2C bus in a wait by setting the interrupt flag to "1", after receiving an acknowledge upon the transmission of one byte when WSEL is set to "0", or immediately after one byte has been transmitted when WSEL is set to "1". Repeat (4) to (6) until a specified number of data elements are transmitted. However, another interrupt occurs upon the reception of an acknowledge, causing the bus to wait, when a NACK is received after the wait is cancelled with WSEL set to "1".

  (7) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

- When transmission FIFO is enabled:

  (1) Set the reserved address to the TDR register as the slave address and write "1" to the MSS bit.

  (2) The interrupt flag (INT) will be set to "1" once the slave address has been transmitted.

  (3) Read from the RDR register to check the reserved address.*

  (4) Write all the data to be transmitted (in case that the transmission FIFO becomes full, write as much until reaching that state) to the TDR register.

  (5) If a NACK is received during transmission, set the interrupt flag (INT) to "1" immediately after the reception to put the I^2C bus in a wait. If all receive an ACK response, set the interrupt flag to "1" after the last byte has been transmitted, according to the WSEL setting, to put the I^2C bus in a wait.

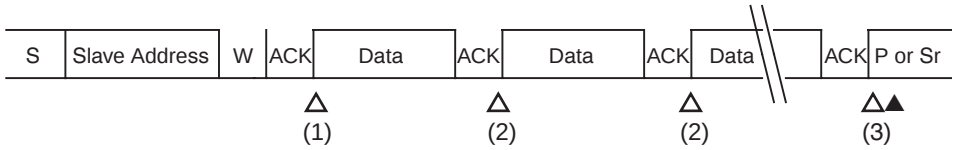
  (6) Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

  *: When the reserved address is a general call address in multi-master operation, it is necessary to confirm whether the device will operate as the master or slave for the next data by setting the ACKE and WSEL bits to "1", if the device may operate as the slave due to the generation of an arbitration lost condition.

<Notes>

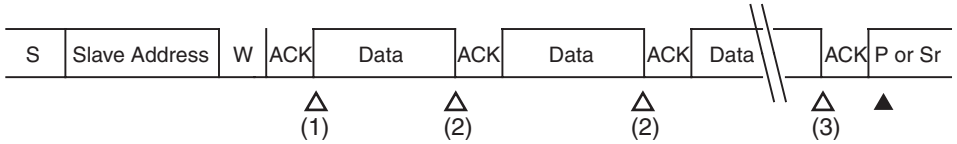
- To modify the IBCR register during transmission or reception, modify it when the interrupt flag (INT) is set to "1".
- When the WSEL bit has been modified, this will be used as a condition for generating the interrupt flag (INT) for the next data.
- If transmission data is written to the TDR register and an ACK response is detected when the TDRE is set to "1" during data transmission, the written data will be transmitted without setting the interrupt flag (INT) to "1".
- If transmission data is written to the TDR register and an ACK is returned when the TDRE is set to "1" during data reception, only RDRF will be set to "1" without setting the interrupt flag (INT) to "1" (when the reception FIFO is enabled, and the amount set in the FBYTE1/FBYTE2 register is received).

Figure 28.20-12 Master Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



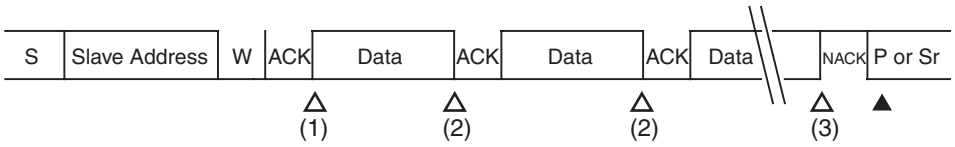
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte + reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-13 Master Transmission Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0, ACK Response)



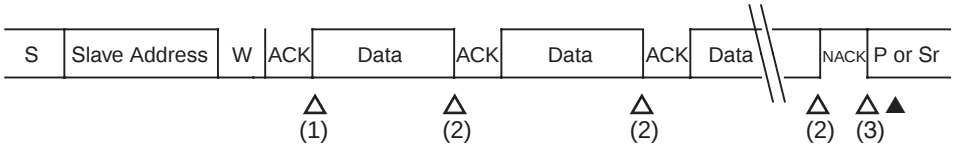
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-14 Master Transmission Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0, NACK Response)



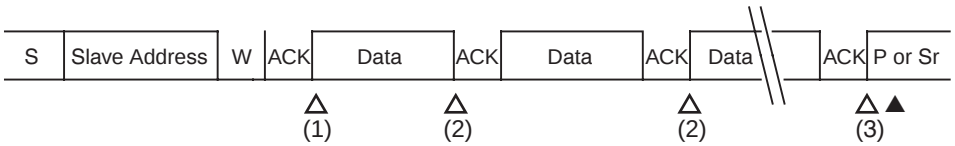
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-15 Master Transmission Interrupt (4) - when FIFO is Disabled (WSEL = 1, RSA = 0, NACK Response in the Middle of Operation)



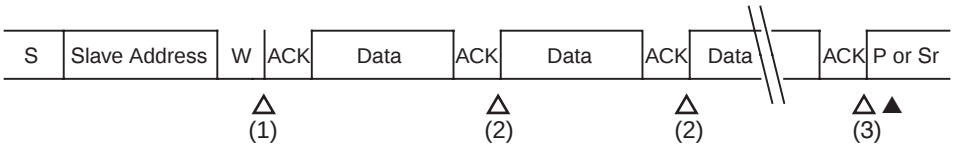
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by NACK response
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-16 Master Transmission Interrupt (5) - when FIFO is Disabled (WSEL = 1 -> 0, RSA = 0, ACK Response)



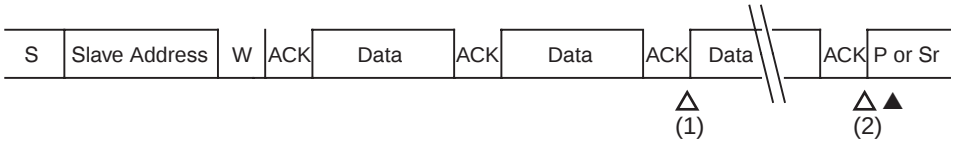
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to the transmission buffer
(2) Interrupt generated by transmission of 1 byte
Writing "0" to WSEL and INT after writing transmission data to the transmission buffer
(3) Interrupt generated by transmission of 1 byte
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-17 Master Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)



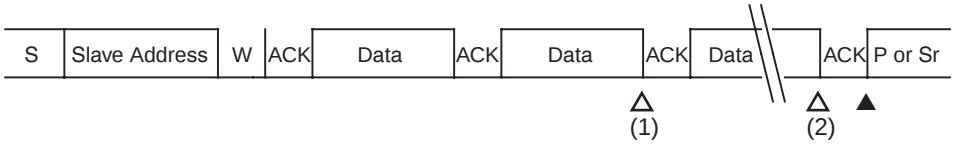
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address (reserved address) + transmission of direction bit + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(2) Interrupt generated by transmission of 1 byte + reception of acknowledge
Writing "0" to INT after writing transmission data to TDR register
(3) Interrupt generated by transmission of 1 byte + reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1
Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-18 Master Transmission Interrupt (7) - when FIFO is Enabled (WSEL = 0, RSA = 0, ACK Response)



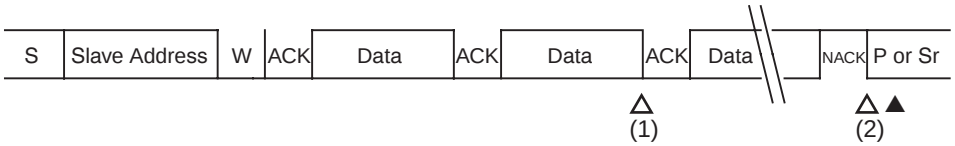
S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
- Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty) + reception of acknowledge
Setting MSS=0 or MSS=1, and SCC=1

Figure 28.20-19 Master Transmission Interrupt (8) - when FIFO is Enabled (WSEL = 1, RSA = 0)



S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by transmission of last byte (transmission FIFO being empty)
Setting MSS=0 or MSS=1, and SCC=1

Figure 28.20-20 Master Transmission Interrupt (9) - when FIFO is Enabled (WSEL = 1, RSA = 0, NACK Response)



S : Start condition
W: Data direction bit (write direction)
P : Stop condition
Sr: Repeated start condition
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by the empty state of transmission FIFO
Writing "0" to INT after writing transmission data to transmission FIFO
(2) Interrupt generated by NACK response
Setting MSS=0 or MSS=1, and SCC=1

■ Master Data Reception

The data transmitted from the slave is received when the data direction bit (R/W) is set to "1".

When FIFO is disabled, the master will generate a wait for reception of each byte if the TDRE bit is set to "1" (INT = 1, RDRF = 1), and an ACK or NACK will be returned by the setting of the ACKE bit in the IBCR register, according to the WSEL bit. When the TDRE bit is set to "0", a wait will not be generated (INT = 0) and the next data will be received if ACK has been selected by the ACKE bit in the IBCR register, or a wait will be generated (INT = 1) if NACK has been selected.

When FIFO is enabled, the RDRF bit will be set if the same number of bytes as a specified number of bytes to be received is received. The interrupt flag is set and puts the I^2C bus in a wait, when the TDRE bit is set to "1". When WSEL is set to "0", setting the TDRE bit to "1" returns a NACK and sets the interrupt flag to "1". When WSEL is set to "1", a wait is generated after the last byte has been received. Therefore, the ACKE bit should be set during that wait to clear the interrupt flag to "0", and then an ACK or NACK should be returned depending on the ACKE setting. Even when a NACK is output, it will be stored as reception data to the reception FIFO.

For interrupt-triggered waits, refer to the following section

Table 28.20-4 WSEL Bit During Master Data Reception

WSEL bit	Operation
0	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after an acknowledge by setting the TDRE bit to "1".
1	A wait is generated in the second or succeeding byte, when the interrupt flag (INT) is set to "1" and SCL is set to "L" after the master transmits 1-byte data by setting the TDRE bit to "1".

An example procedure for receiving data from the slave is shown below.

- When reception FIFO is disabled:
  (1) Set the slave address (including the data direction bit) to the TDR register and write "1" to the MSS bit.
  (2) An ACK will be received after the slave address is transmitted, and then the interrupt flag (INT) will be set to "1".
  (3) Update the WSEL bit and write "0" to the interrupt flag (INT) to cancel a wait for the I^2C bus.

(4) Put the I^2C bus in a wait by setting the interrupt flag to "1", after transmitting an acknowledge upon the reception of one byte when WSEL is set to "0", or immediately after one byte has been received when WSEL is set to "1". Repeat (2) to (4) until a specified number of data elements are received.

(5) Output a NACK after the reception of the last data, and set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

- When transmission/reception FIFO is enabled:

(1) Set the number of receptions to the FBYTE1/FBYTE2 register.

(2) Write the slave address (including the data direction bit) and dummy data for the number of receptions to the TDR register.

(3) Write "1" to the MSS bit.

(4) ACK will be returned and reception will continue as long as the TDRE bit is set to "0". RDRF is set to "1" once the amount set in FBYTE1/FBYTE2 is received during that reception. When RDRF is set to "1", the RDR register is read.

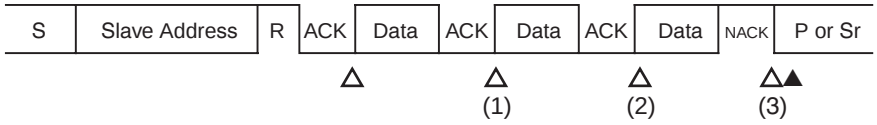
(5) When the TDRE bit is set to "1", the interrupt flag will be set to "1" to put the I^2C bus in a wait, after the output of a NACK if WSEL is set to "0", or immediately after the reception of one byte if WSEL is set to "1".

(6) The ACKE bit should be set to "0" when WSEL is set to "1", or the ACKE bit does not have to be set when WSEL is set to "0". Set the MSS bit to "0" or the SCC bit to "1" to generate a stop condition or a repeated start condition.

<Notes>

- During enabling the detection of 7-bit slave address (ISBA:SAEN=1), it is prohibited to specify 7-bit slave address in master mode.
- An acknowledge will be output to handle the next data according to the setting of the ACKE bit, even if an overrun error occurs when TDRE is set to "0".
- Modify the IBCR register during transmission/reception, if necessary, when the interrupt flag (INT) is set to "1".
- In master reception, the next data will be received with the interrupt flag (INT) still set to "0", if the TDRE bit is set to "0" when dummy data is written to the TDR register and the interrupt flag (INT) is set to "1".
- If data is received when the reception FIFO has been enabled and WSEL is set to "0", the RDRF bit will be set to "1" upon the reception of the last bit and the interrupt flag (INT) will be set to "1" after an ACK is transmitted.

Figure 28.20-21 Master Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



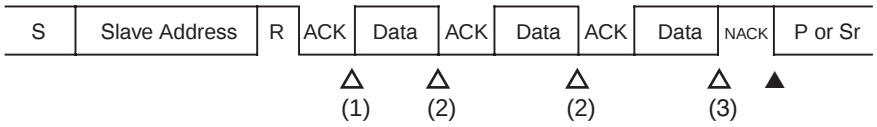
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
- Interrupt cleared to "0" by writing "0" to INT

(2) Interrupt generated by reception of 1 byte + transmission of acknowledge
- Setting ACKE to "0" and writing "0" to INT after reading reception data

(3) Interrupt generated by reception of 1 byte + transmission of acknowledge
- Setting MSS=0 or MSS=1, and SCC=1

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-22 Master Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by transmission of slave address + transmission of direction bit + reception of acknowledge
- Interrupt cleared to "0" by writing "0" to INT

(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reading reception data

(3) Interrupt generated by reception of 1 byte
- Setting ACKE=0, and then setting MSS=0 or MSS=1, and SCC=1 after reading reception data

Note: The TDRE bit is set to "1" when the interrupt flag (INT) is generated.

Figure 28.20-23 Master Reception Interrupt (3) - when FIFO is Enabled (WSEL = 0, ACKE = 0, RSA = 0)

△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by TDRE=1
- Setting MSS=0 or MSS=1, and SCC=1 after reading all data from reception FIFO

Figure 28.20-24 Master Reception Interrupt (4) - when FIFO is Enabled (WSEL = 1, RSA = 0)

△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by TDRE=1
- Setting ACKE=0, MSS=0 or MSS=1, and SCC=1 after reading all data from reception FIFO

■ Arbitration Lost Condition

When a master receives data which is different from the transmitted data due to a data collision with the data from another master, this is determined as an arbitration lost condition. Consequently, the MSS bit is set to "0" and the AL bit to "1" to allow the device to operate in slave mode.

The AL bit can be cleared to "0" under the following conditions.

- "1" is written to the MSS bit.
- "0" is written to the INT bit.
- "0" is written to the SPC bit when the AL and SPC bits are set to "1".
- The I^2C interface is disabled (EN bit = 0).

When an arbitration lost condition occurs, the interrupt flag (INT) is set to "1" and the SCL of the I^2C bus is set to "L", according to the setting of WSEL.

■ Wait in Master Mode

If the device is not operating in slave mode when the MSS bit is set to "1" with the BB bit set to "1", the master mode will be put in a wait as long as the BB bit remains set to "1". It will transmit a start condition once the BB bit becomes "0". The MSS and ACT bits can be used to determine whether the master mode is in a wait or not (MSS = 1, ACT = 0: wait state). To allow the device to operate in slave mode after the MSS bit is set to "1", set AL = 1, MSS = 0, and ACT = 1.

28.20.3 Slave Mode

In slave mode, the device detects a (repeated) start condition and returns an ACK when the combination of the ISBA and ISMK registers matches the received address, in order to operate in slave mode.

■ Slave Address Match Detection

When a (repeated) start condition is detected, the next 7-bit data is received as an address. Each bit of the ISBA register is compared with the corresponding bit of the received address for the bits which are set to "1" in the ISMK register. An ACK will be output if there is a match.

Table 28.20-5 Operation Immediately after Output of Acknowledge for Slave Address

Transmission FIFO	Reception FIFO	Transmission FIFO status	Reception FIFO status	Data direction bit (R/W)	Operation immediately after acknowledge	
					Acknowledge = ACK	Acknowledge = NACK
Disabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Disabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	
Enabled	Disabled	-	-	0	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	The INT bit remains set to "0", causing no wait.
				1		
Enabled	Enabled	-	No data contained	0	The INT bit remains set to "0", causing no wait.	The INT bit remains set to "0", causing no wait.
			Data contained		The INT bit is set to "1", causing a wait.	
			-	1	The INT bit is set to "1", causing a wait, when the TDRE bit is set to "1". The INT bit remains set to "0", causing no wait, when the TDRE bit is set to "0".	

- Reserved address detection

  When the first byte matches a reserved address ("$0000XXXX_B$" or "$1111XXXX_B$"), the INT bit is set to "1" to put the I^2C bus in a wait upon the reception of data from the 8th bit, whether or not the transmission/reception FIFO is enabled. At this point, ACKE is set to "1" and the INT bit is cleared when allowing the device to operate as a slave. The device will then start slave operation. When ACKE is set to "0", the device does not operate as a slave after the output of an acknowledge.

■ Data Direction Bit

The data direction bit, which determines data transmission or reception, is received after an address is received. When this bit is set to"0", this indicates transmission from the master, therefore, as a slave, the device will receive data.

■ Slave Reception

Reception is performed in slave mode when there is a slave address match and the data direction bit is set to "0". An example procedure for reception in slave mode is shown below.

- When reception FIFO is disabled:

  (1) Set the interrupt flag (INT) to "1" to put the I^2C bus in a wait after an ACK is transmitted. When the MSS, ACT and FBT bits determine that the interrupt is caused by a slave address match, set the ACKE bit to "1" and write "0" to the interrupt flag (INT) to cancel the I^2C bus wait. (Refer to Table 28.20-5.)

  (2) After 1-byte data is received, set the interrupt flag (INT) to "1" according to the WSEL setting to put the I^2C bus in a wait.

  (3) Read the data received from the RDR register, set the ACKE bit and then write "0" to the interrupt flag (INT) to cancel the I^2C bus wait.

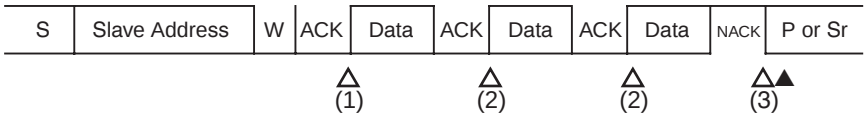
  (4) Repeat (2) and (3) until a stop condition or a repeated start condition is detected.

- When reception FIFO is enabled:

  (1) The interrupt flag (INT) is set to "1" to put the I^2C bus in a wait when a NACK is detected or the reception FIFO becomes full. When a stop condition or a repeated start condition is detected, the SPC and RSC bits are set to "1" but not the interrupt flag (INT) (no I^2C bus wait). The reception FIFO sets the RDRF bit to "1" when the value set in the FBYTE1/FBYTE2 register matches the number of data elements received. At this point, a reception interrupt will occur if the RIE bit has been set to "1".

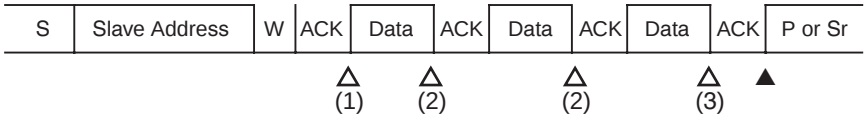
  (2) When the interrupt flag (INT) is set to "1", read the data received from the RDR register. After reading all the data, write "0" to the interrupt flag to cancel the I^2C bus wait. Read all the data received from the RDR register and clear the SPC or RSC bit to "0", if a stop condition or a repeated start condition is detected.

Figure 28.20-25 Slave Reception Interrupt (1) - when FIFO is Disabled (WSEL = 0, RSA = 0)



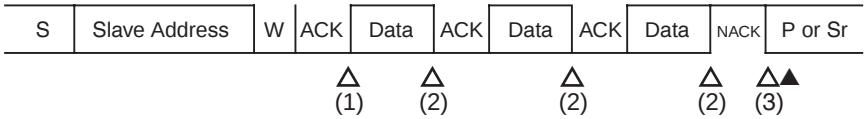
△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte + ACK response
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by reception of 1 byte + NACK response
- Writing "0" to INT after reception data is read from reception buffer

Figure 28.20-26 Slave Reception Interrupt (2) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer

Figure 28.20-27 Slave Reception Interrupt (3) - when FIFO is Disabled (WSEL = 1, RSA = 0)



△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by ACK output due to slave address match
- Writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte
- Writing "0" to INT after reception data is read from reception buffer
(3) Interrupt generated by NACK response
- Writing "0" to INT

Figure 28.20-28 Slave Reception Interrupt (4) - when Reception FIFO is Enabled (RSA = 0)

△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by detection of stop condition or repeated start condition
- Reading all data from reception FIFO

Figure 28.20-29 Slave Reception Interrupt (5) - when Reception FIFO is Enabled (RSA = 0)

△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by reception FIFO being full
- Reading all data from reception FIFO and writing "0" to INT

Figure 28.20-30 Slave Reception Interrupt (6) - when FIFO is Disabled (WSEL = 0, RSA = 1)

△: Interrupt by INTE=1
▲: Interrupt by CNDE=1
(1) Interrupt generated by reserved address ("0000XXXX$_B$" or "1111XXXX$_B$") match
- Reading reception data and writing "1" to ACKE and "0" to INT
(2) Interrupt generated by reception of 1 byte + output of acknowledge
- Writing "0" to INT
(3) Interrupt generated by reception of 1 byte + output of acknowledge
- Interrupt by writing "0" to INT

■ Slave Transmission

Transmission is performed in slave mode when there is a slave address match and the data direction bit is set to "1". When FIFO is disabled, a wait is generated by setting the interrupt flag (INT) to "1" after transmitting one byte or after returning an acknowledge, depending on the WSEL setting (see Table 28.20-5).

The RACK bit can be used to confirm the acknowledge output from the master. It indicates the end of the data reception, determining whether or not the master succeeded in the reception at a time of NACK response. An interrupt will occur to generate a wait if a NACK is detected when WSEL is set to "1".

28.20.4 Bus Error

A case where a stop condition or a (repeated) start condition is detected during data transmission/reception on the I^2C bus is handled as a bus error.

■ Conditions for the Occurrence of Bus Errors

A bus error sets the BER bit to "1" under the following conditions.

- A (repeated) start condition or a stop condition is detected during the transfer of the first byte.
- A (repeated) start condition or a stop condition is detected in the 2nd bit - 9th (acknowledge) bit of data.

■ Bus Error Operation

Check the BER bit when transmission/reception sets the interrupt flag (INT) to "1". If the BER bit is set to "1", the error must be treated. The BER bit is cleared when "0" is written to the INT bit.

Although a bus error sets the INT bit to "1", the I^2C bus does not enter a wait state with SCL set to "L".

28.21 Dedicated Baud Rate Generator

The dedicated baud rate generator sets a serial clock frequency.

■ Baud Rate Selection

● Baud rate achieved by dividing the internal clock using the dedicated baud rate generator (reload counter)

There are two internal reload counters, and both support the transmission/reception serial clock. The baud rate can be selected via the 15-bit reload value determined by the baud rate generator registers 1, 0 (BGR1, BGR0).

The reload counter divides the internal clock, according to the set value.

■ Calculating the Baud Rate

The two 15-bit reload counters are set by the baud rate generator registers 1, 0 (BGR1, BGR0).

The following formula should be used to calculate a baud rate.

(1) Reload value:

$V = \phi / b - 1$

V: Reload value b: Baud rate ϕ: Peripheral clock (PCLK) frequency

Note that the set baud rate may not be generated depending on the SCL rising time of the I^2C bus. In that case, the reload value must be adjusted.

(2) Example of calculation:

If the peripheral clock (PCLK) is 16MHz and the baud rate is 400kbps, the reload value will be:

Reload value:

$V = (16 \times 1000000)/400000 - 1 = 39$
As a result, the baud rate is:
$b = (16 \times 1000000)/(38+2) = 400$ kbps

<Notes>

- Use 16-bit access to write to the baud rate generator registers 1, 0 (BGR1, BGR0).
- Set the baud rate generator registers when the EN bit in the ISMK register is "0".
- Use the peripheral clock (PCLK) at 8 MHz or higher in operation mode 4 (I^2C mode). It is prohibited to set the baud rate generator to higher than 400kbps.
- The reload counter stops when the reload value is set to "0".

■ Reload Values and Baud Rates for Different Peripheral Clock (PCLK) Frequencies

Table 28.21-1 Reload Values and Baud Rates

Baud rate [bps]	8 MHz Reload value	10 MHz Reload value	16 MHz Reload value	20 MHz Reload value	24 MHz Reload value	32MHz Reload value
400000	19	24	39	49	59	79
200000	39	49	79	99	119	159
100000	79	99	159	199	239	319

These numerical values are based on the SCL rising time of I^2C bus set to "0". If the rising is slower, the actual baud rates should also be slower than the numerical values above.

■ Functions of Reload Counters

Structured in a 15-bit register configuration based on a reload value, these counters generate a transmission/reception clock from the internal clock. In addition, the count value of the transmission reload counter can be read from the baud rate generator registers 1, 0 (BGR1, BGR0).

■ Starting a Count

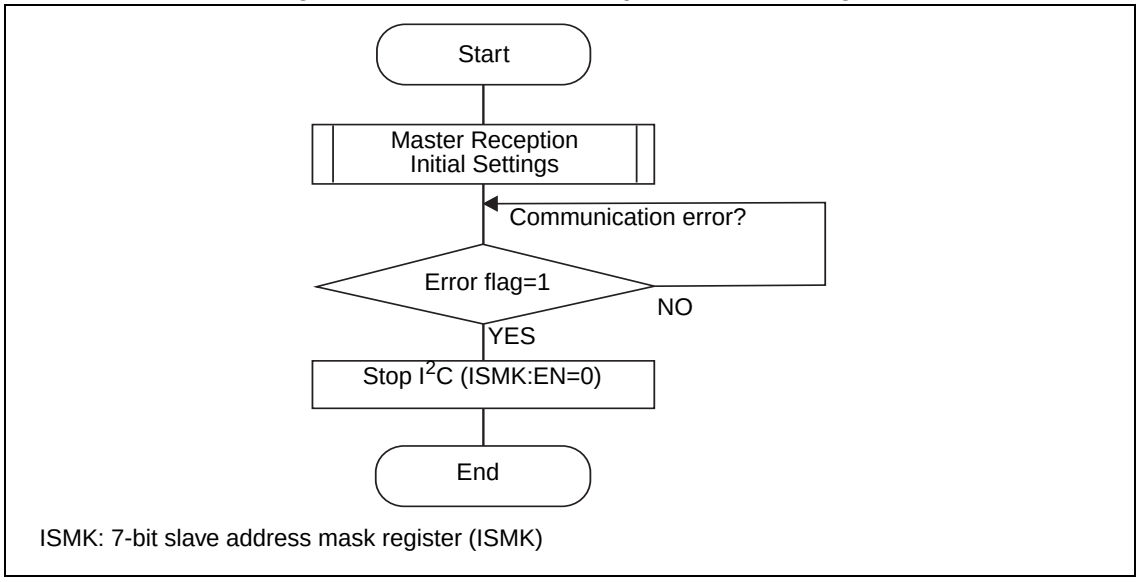
The reload counter starts a count when a reload value is written to the baud rate generator registers 1, 0 (BGR1, BGR0).

28.21.1 Example of I²C Flowcharts

Below are some example flowcharts for I²C communication.

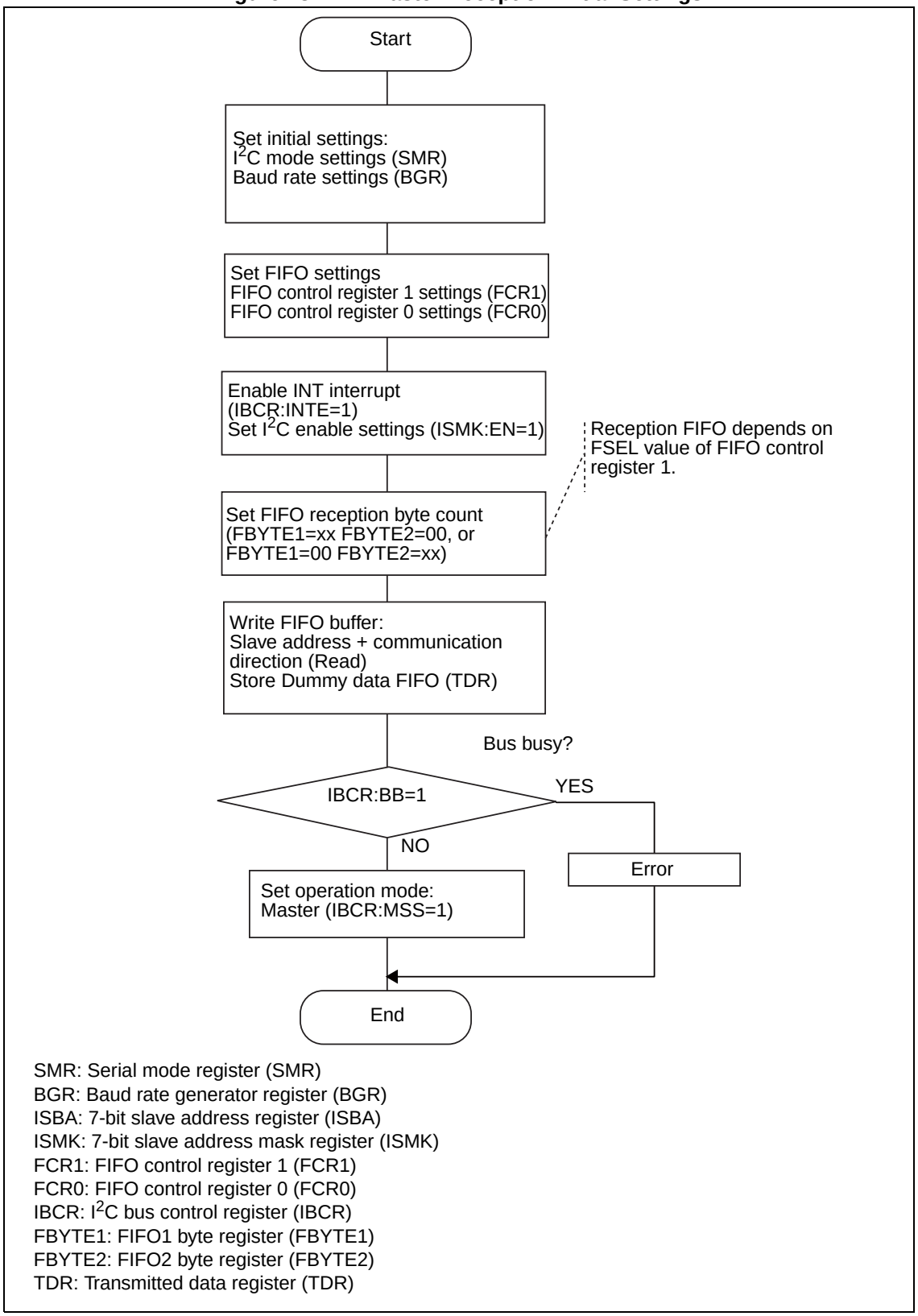
■ I²C Master Reception/ Slave Transmission FIFO Communication Flow

Figure 28.21-1 Master Reception Main Settings



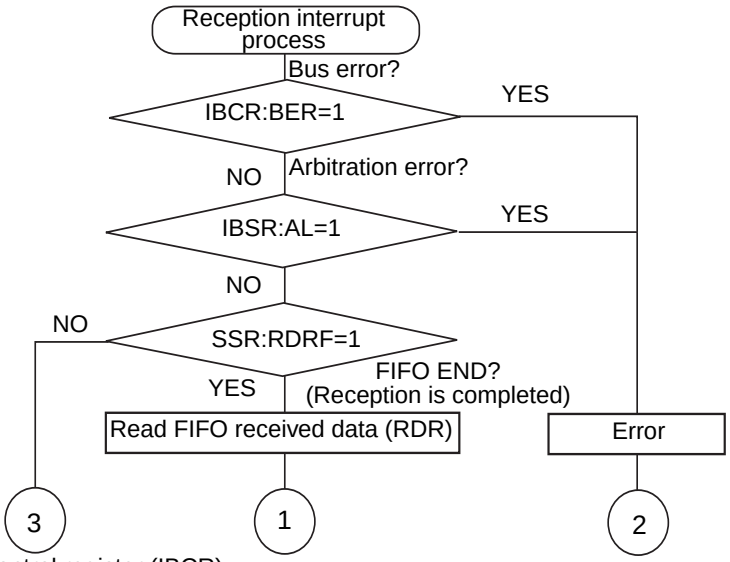
ISMK: 7-bit slave address mask register (ISMK)

Figure 28.21-2 Master Reception Initial Settings



SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)
FBYTE1: FIFO1 byte register (FBYTE1)
FBYTE2: FIFO2 byte register (FBYTE2)
TDR: Transmitted data register (TDR)

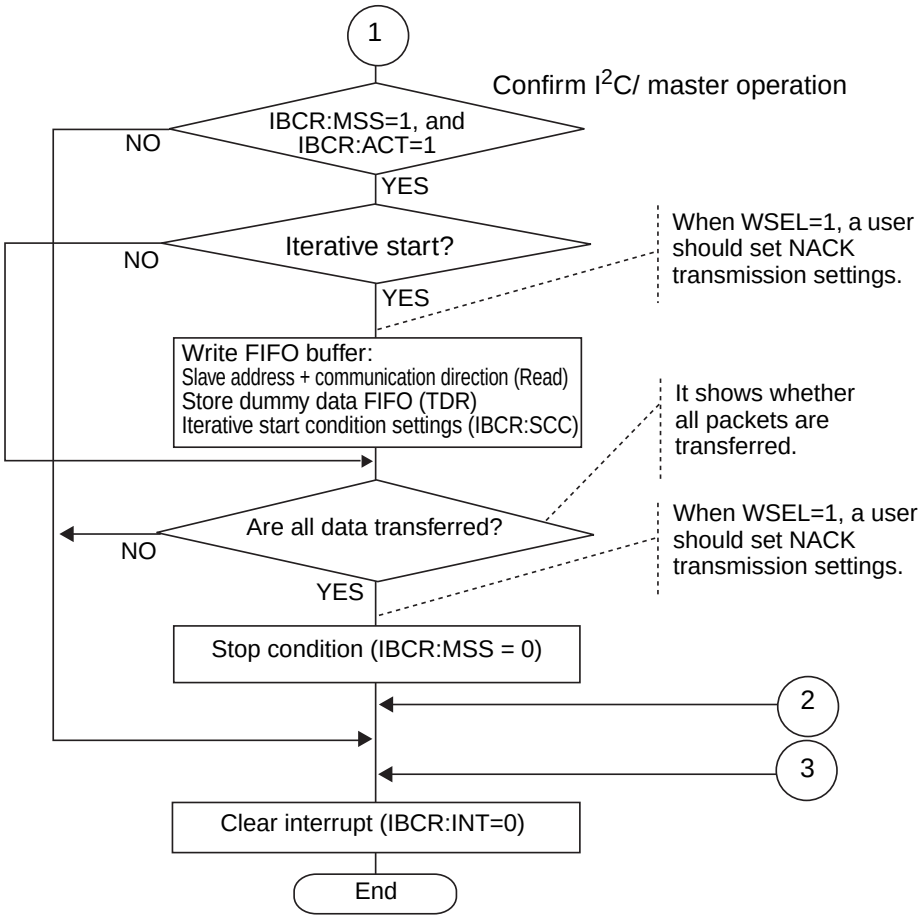
Figure 28.21-3 Master Reception Interrupt Process



IBCR: I^{2}C bus control register (IBCR)
IBSR: I^{2}C bus status register (IBSR)
SSR: serial status register (SSR)
RDR: received data register (RDR)

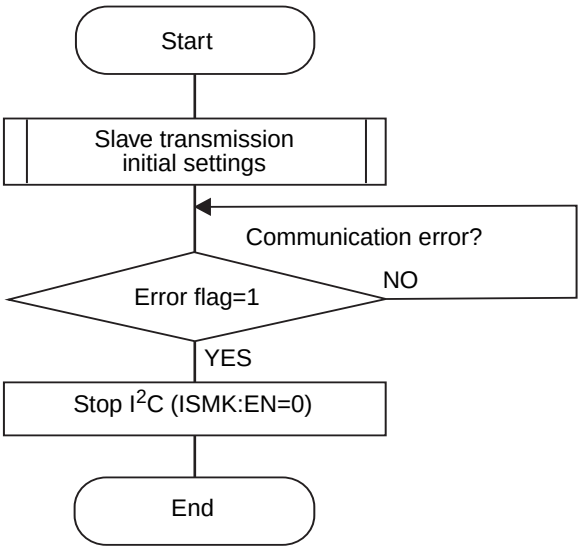
* For actual error handling, please judge each status error flag and handle each error by considering your systems.

* For final ACK process when receiving FIFO continuously, a wait will be generated after receiving the final data, if setting WSEL=1. Please respond with NACK by setting IBCR.RACK=1 during this wait.



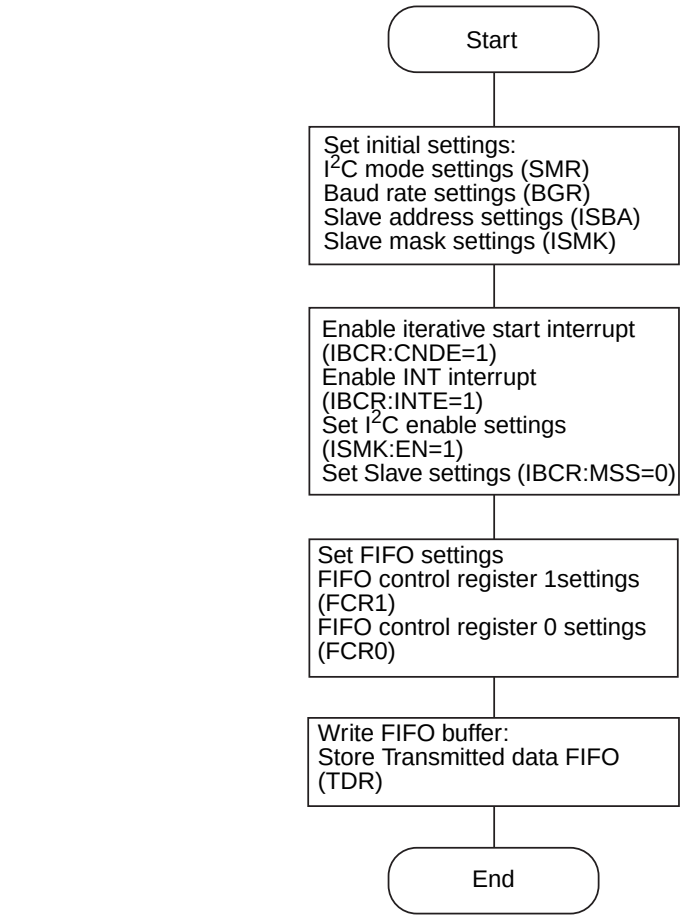
IBCR: I^{2}C bus control register (IBCR)
TDR: Transmitted data register (TDR)

Figure 28.21-4 Slave Transmission Main Settings



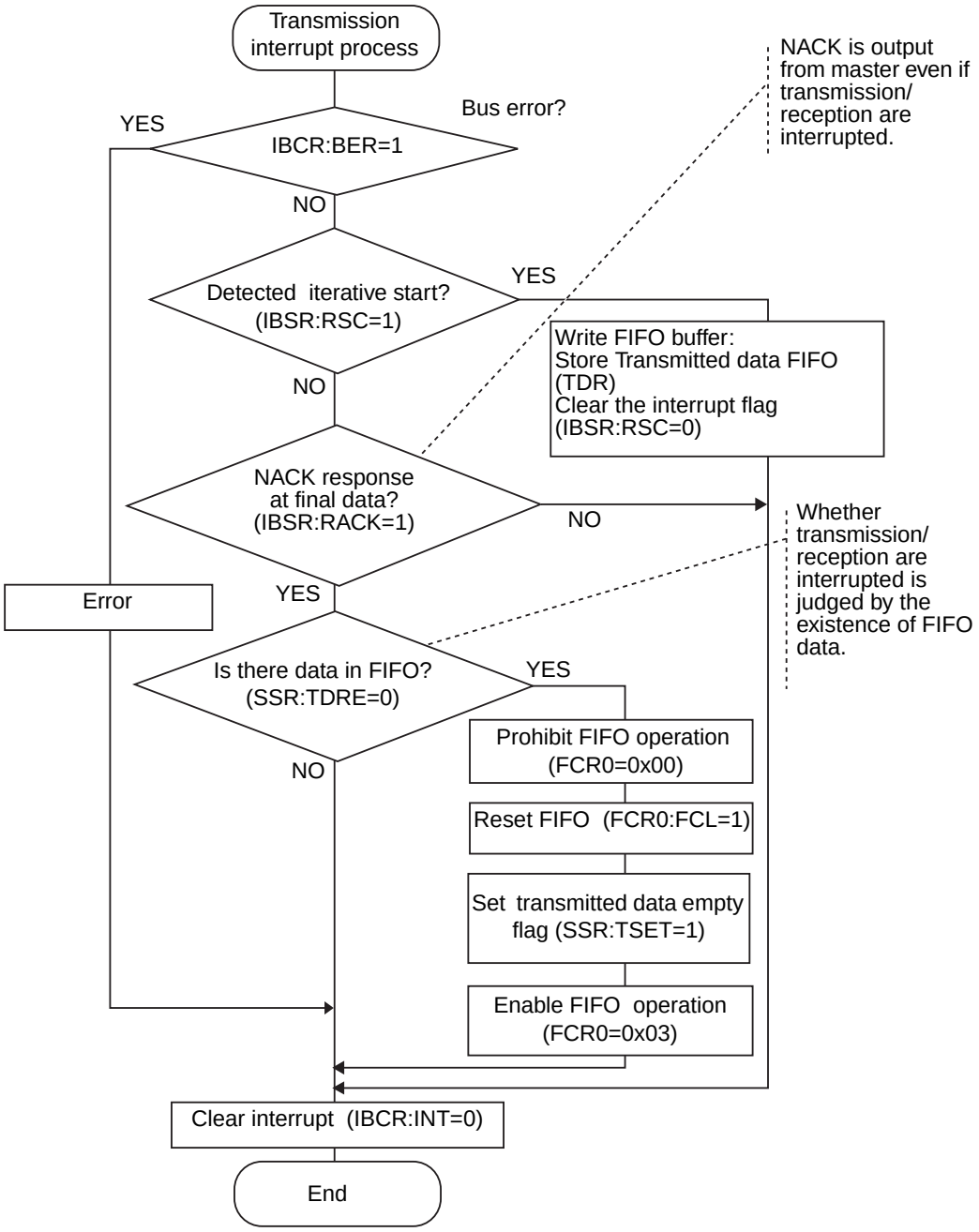
ISMK: 7-bit slave address mask register (ISMK)

Figure 28.21-5 Slave Transmission Initial Settings



SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
IBCR: I^2C bus control register (IBCR)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
TDR: Transmitted data register (TDR)

Figure 28.21-6 Slave Transmission Interrupt Process

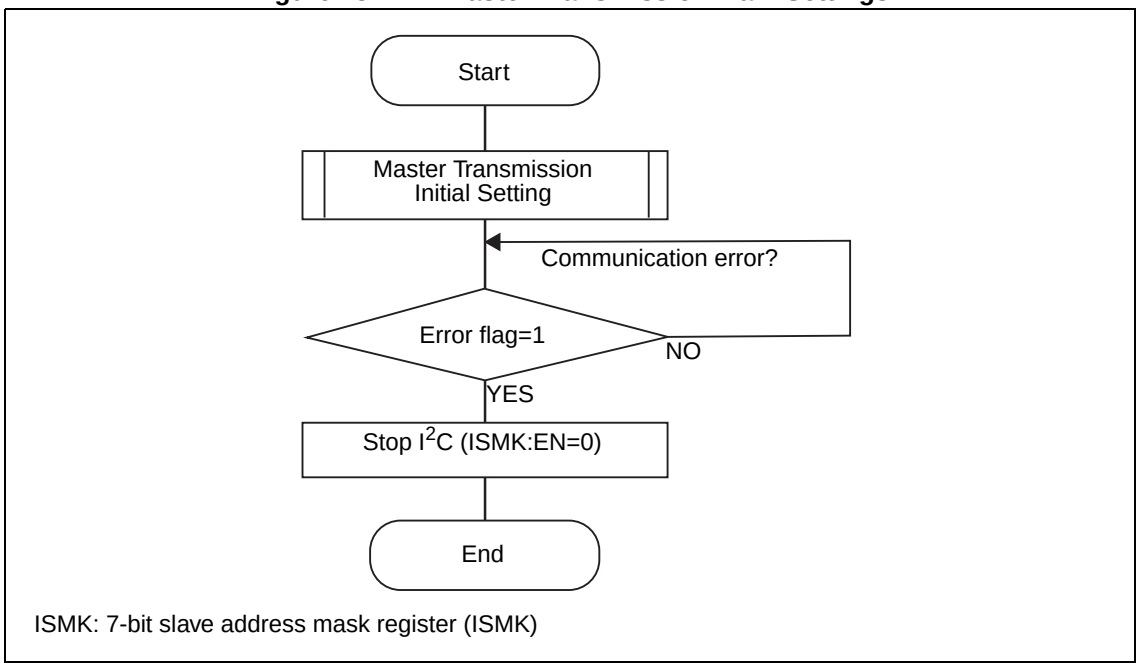


IBSR: I^2C bus status register (IBSR)
TDR: Transmitted data register (TDR)
SSR: Serial status register (SSR)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

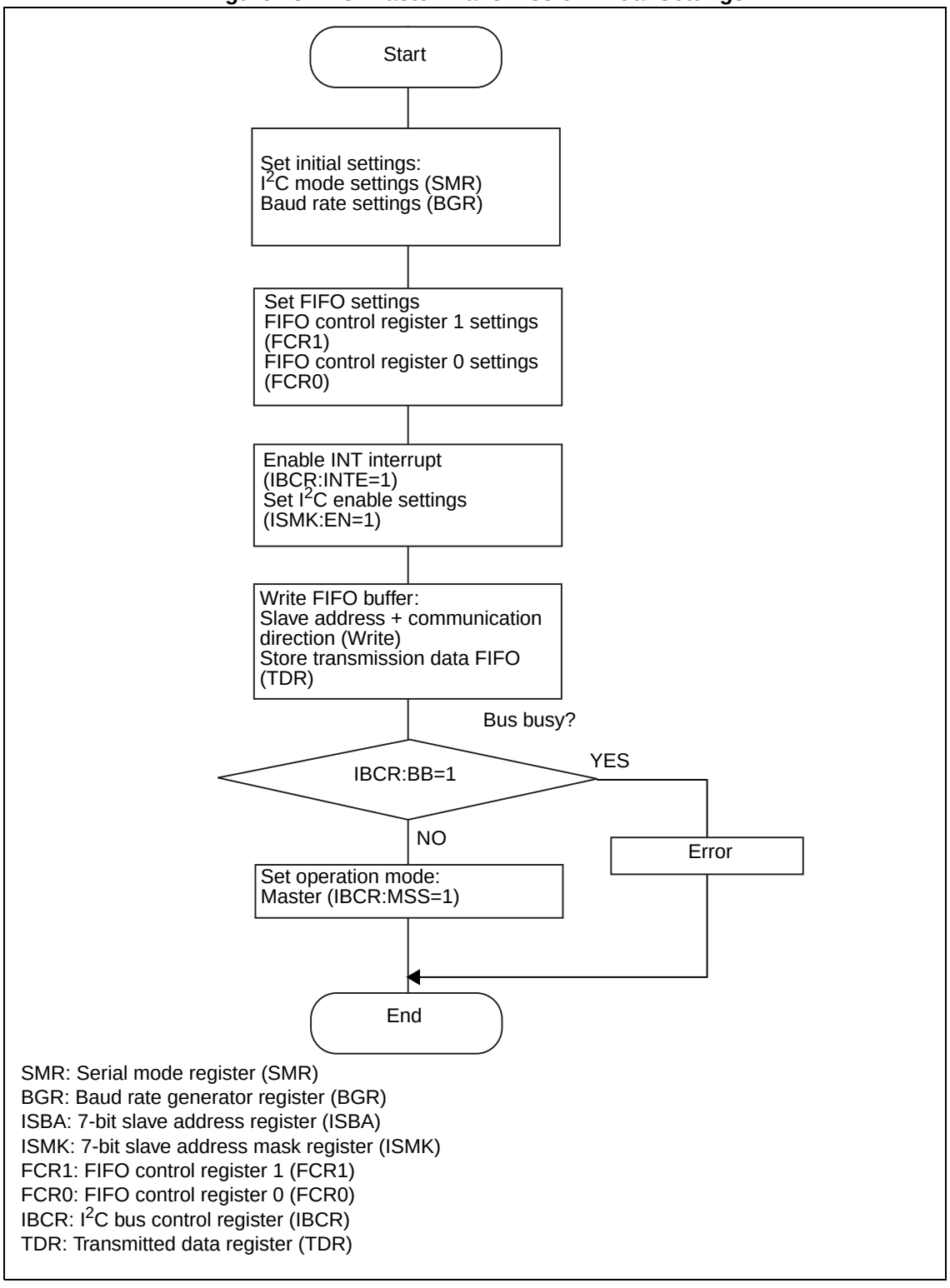
■ I²C Master Transmission/ Slave Reception FIFO Communication Flow

Figure 28.21-7 Master Transmission Main Settings



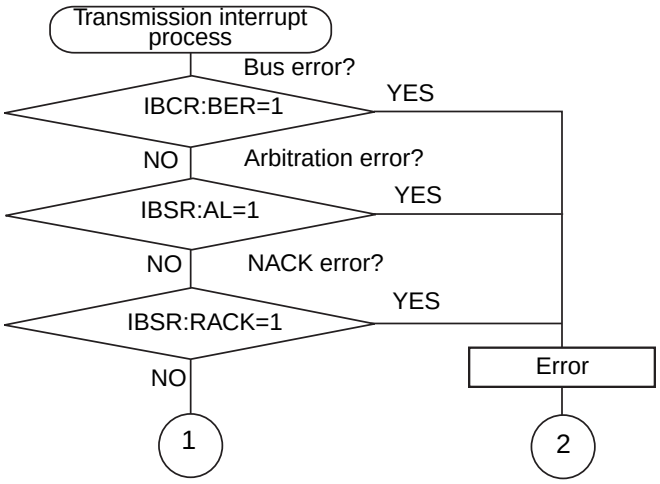
ISMK: 7-bit slave address mask register (ISMK)

Figure 28.21-8 Master Transmission Initial Settings



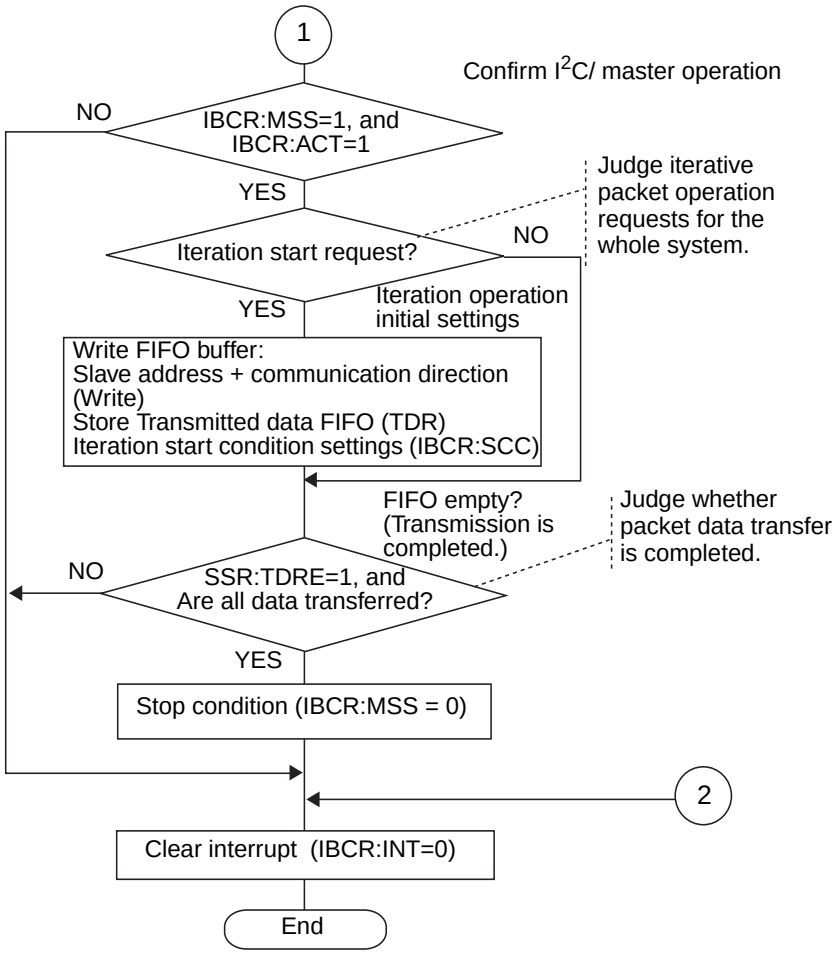
SMR: Serial mode register (SMR)
BGR: Baud rate generator register (BGR)
ISBA: 7-bit slave address register (ISBA)
ISMK: 7-bit slave address mask register (ISMK)
FCR1: FIFO control register 1 (FCR1)
FCR0: FIFO control register 0 (FCR0)
IBCR: I^2C bus control register (IBCR)
TDR: Transmitted data register (TDR)

Figure 28.21-9 Master Transmission Interrupt Process



IBCR: I^2C bus control register (IBCR)
IBSR: I^2C bus status register (IBSR)

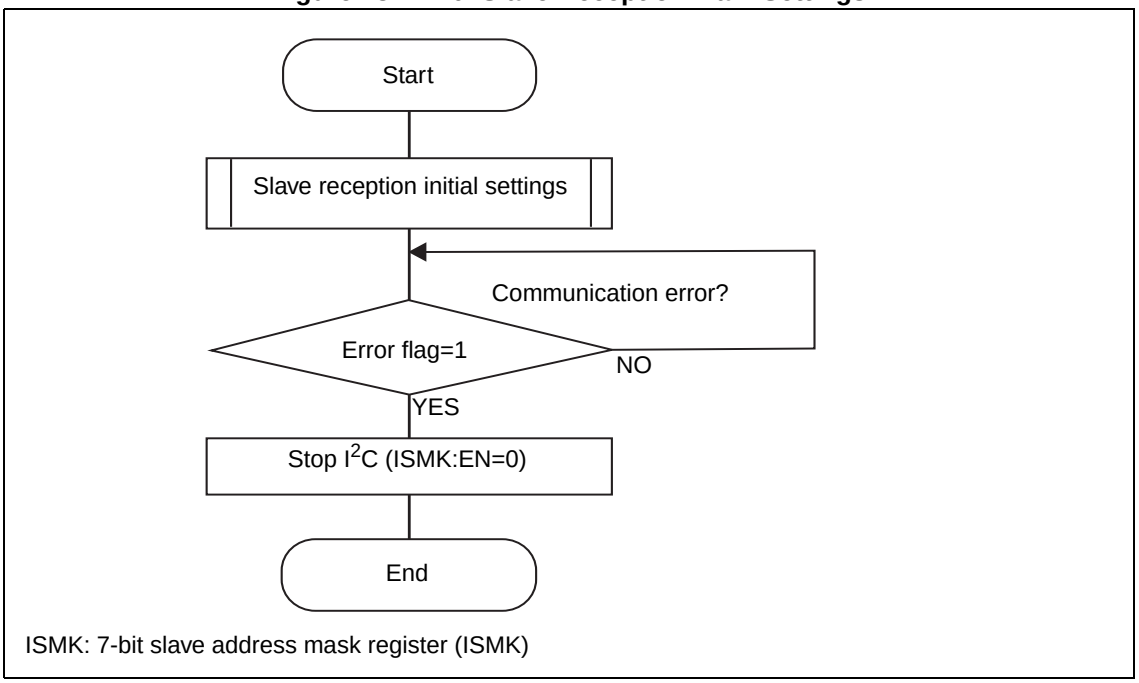
* For actual error handling, please judge each status error flag and handle each error by considering your systems.



IBCR: I^2C bus control register (IBCR)
TDR: Transmitted data register (TDR)
SSR: Serial status register (SSR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

Figure 28.21-10 Slave Reception Main Settings



ISMK: 7-bit slave address mask register (ISMK)

Figure 28.21-11 Slave Reception Initial Settings

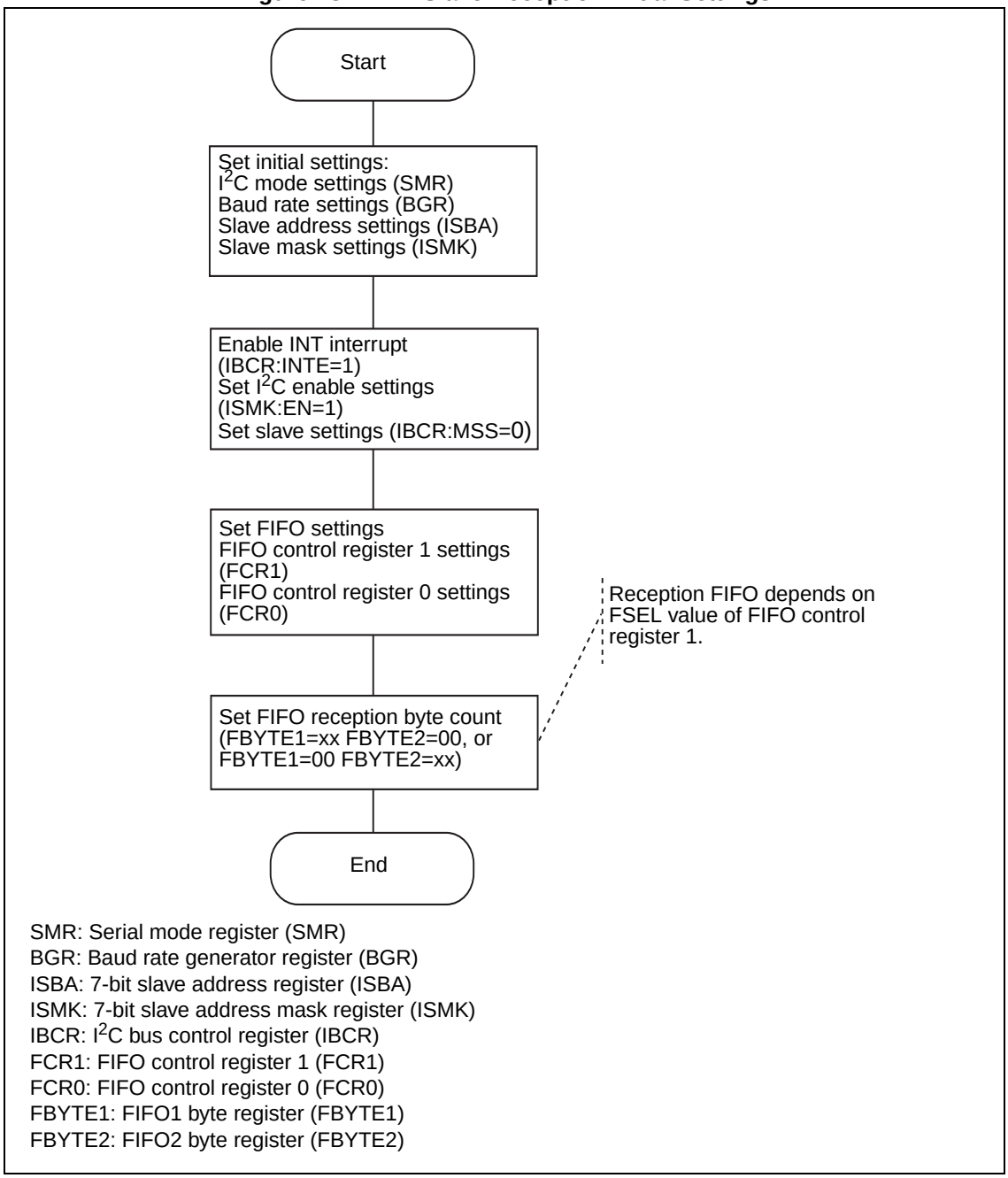
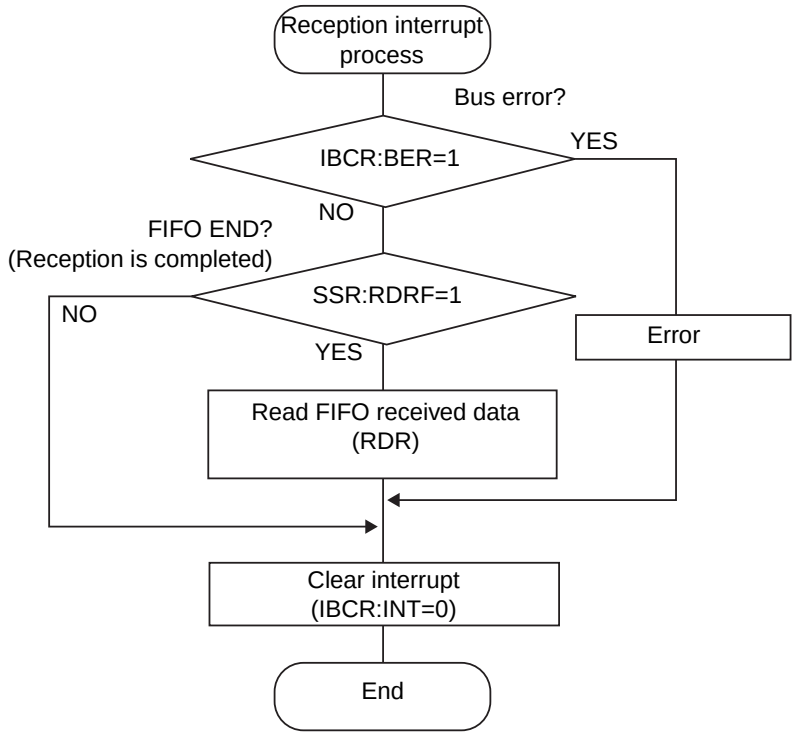


Figure 28.21-12 Slave Reception Interrupt Process



IBSR: I^2C bus status register (IBSR)
SSR: Serial status register (SSR)
RDR: Received data register (RDR)
IBCR: I^2C bus control register (IBCR)

* For actual error handling, please judge each status error flag and handle each error by considering your systems.

CHAPTER 29 USB Clock Generating Part

This chapter explains the functions and operations of the USB clock generating part.

29.1 Overview

29.2 Configuration

29.3 Registers

29.4 Explanation of Operations and Setting Procedure Examples

29.1 Overview

The USB clock generates the clock used for USB device operations.

■ Overview

The USB clock generating part has the following functions:

- Enabling/Stopping of USB clock output
- PLL macro oscillation clock division
- USB clock selection
- Stopping of the USB clock in standby mode (watch or stop mode)

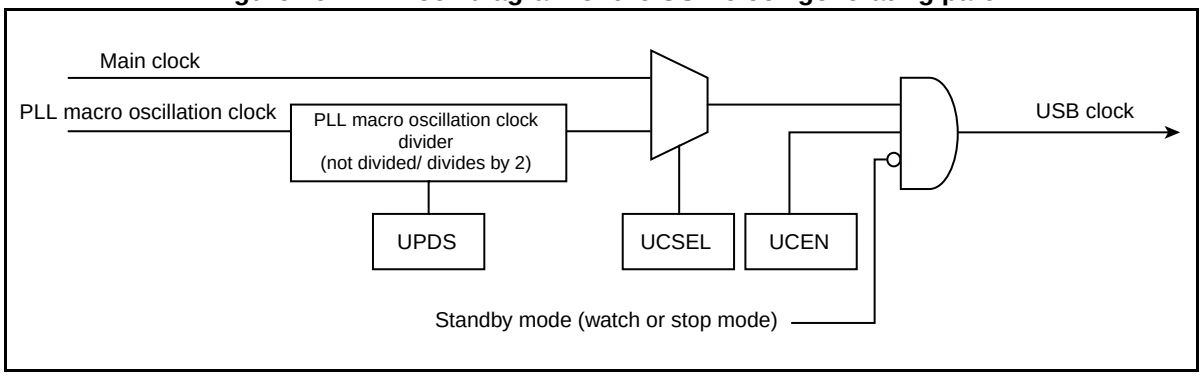
29.2 Configuration

This section explains the configuration of the USB clock generating part.

■ Block diagram of the USB clock generating part

Figure 29.2-1 is a block diagram of the USB clock generating part.

Figure 29.2-1 Block diagram of the USB clock generating part



29.3 Registers

This section explains the configuration and functions of registers used with the USB clock generating part.

■ Registers of the USB clock generating part

Table 29.3-1 lists the registers of the USB clock generating part.

Table 29.3-1 Registers of the USB clock generating part

Abbreviated Register Name	Register Name	Reference
UCCR	USB clock configuration register	29.3.1

29.3.1 USB Clock Configuration Register (UCCR)

This register controls the USB clock.

Figure 29.3-1 shows the bit configuration of the USB clock configuration register (UCCR).

Figure 29.3-1 Bit configuration of the USB clock configuration register (UCCR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	UPDS	UCSEL	UCEN
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	X	X	X	X	X	0	0	1

R/W: Read/Write

X: Undefined

[bit7 to bit3]: Reserved bits

In case of writing	Write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2]: UPDS (PLL macro oscillation clock division select bit)

This bit selects whether to divide the PLL macro oscillation clock.

Written Value	Explanation
0	Sets no division.
1	Sets that the clock be divided by 2.

<Notes>

- If USB clock output is enabled (UCEN=1), this bit cannot be changed.
- This bit and the UCEN bit must not be changed at the same time.

[bit1]: UCSEL (USB clock select bit)

This bit specifies the main clock or PLL macro oscillation clock as the USB clock.

Written Value	Explanation
0	Main clock
1	PLL macro oscillation clock

<Notes>

- If USB clock output is enabled (UCEN=1), this bit cannot be changed.
- This bit and the UCEN bit must not changed at the same time.

[bit0]: UCEN (USB clock output enable bit)

This bit controls USB clock output.

Written Value	Explanation
0	Disables USB clock output.
1	Enables USB clock output.

<Note>

This bit and the UPDS bit or UCSEL bit must not be changed at the same time.

29.4 Explanation of Operations and Setting Procedure Examples

This section explains USB clock generation control. The section also provides a setting procedure example.

■ Overview

The USB clock generation control part requires that operations be performed for the following:

- Register settings
- Main clock selection
- PLL macro oscillation clock selection

■ Register settings

To set the USB clock configuration register (UCCR) to enable USB clock output, follow the procedure below.

1. Disable USB clock output (UCEN=0).
2. Set the UPDS bit (PLL macro oscillation clock division select bit) and UCSEL bit (USB clock select bit).
3. Enable USB clock output (UCEN=1).

<Notes>

- Do not perform steps 1 and 2 or steps 2 and 3 at the same time.
- Make these settings while the USBEN bit of the USB enable register (USBEN) is "0".

■ Main clock selection

Writing "0" to the UCSEL bit specifies the main clock as the USB clock. When specified, the main clock must be oscillating at 48 MHz.

■ PLL macro oscillation clock selection

Writing "1" to the UCSEL bit specifies the PLL macro oscillation clock as the USB clock. When specified, the PLL macro oscillation clock must have a frequency of 48 MHz or 96 MHz. If the frequency of the PLL macro oscillation clock is 48 MHz, set the PLL macro oscillation clock to no division (UPDS=0). If the PLL macro oscillation clock is 96 MHz, set the PLL macro oscillation clock to be divided by 2 (UPDS=1). Table 29.4-1 shows the relationship.

Table 29.4-1 Relationship between the number for the USB clock and the UPDS/UCSEL bit

	UPDS	UCSEL
Main clock frequency of 48 MHz	-	0
PLL macro oscillation clock frequency of 48 MHz	0	1
PLL macro oscillation clock frequency of 96 MHz	1	1

<Note>

The USB clock stops in standby mode (watch or stop mode).

CHAPTER 30 DMA Transfer Request Selector

This chapter explains the functions and operations of the DMA Transfer Request selector.

30.1 Overview
30.2 Configuration
30.3 Registers

30.1 Overview

If DREQ is used as a DMA transfer request, the DMA Transfer Request selector passes DREQ from the USB function with Mini-HOST, and the slave interface to the DMA controller (DMAC). It also returns DACK from the DMA controller (DMAC) to the appropriate macro.

■ Overview

The DMA Transfer Request selector passes DREQ from the USB function with Mini-HOST, and the slave interface to the appropriate channel of the DMA controller (DMAC).

It also returns DACK from the DMA controller (DMAC) to the appropriate macro determined from a comparison with the used channel signal for a connection to the corresponding DACK pin. If the USB function with Mini-HOST is used, an acknowledgement corresponding to the last transfer is used as a DMA transfer end signal for the connection.

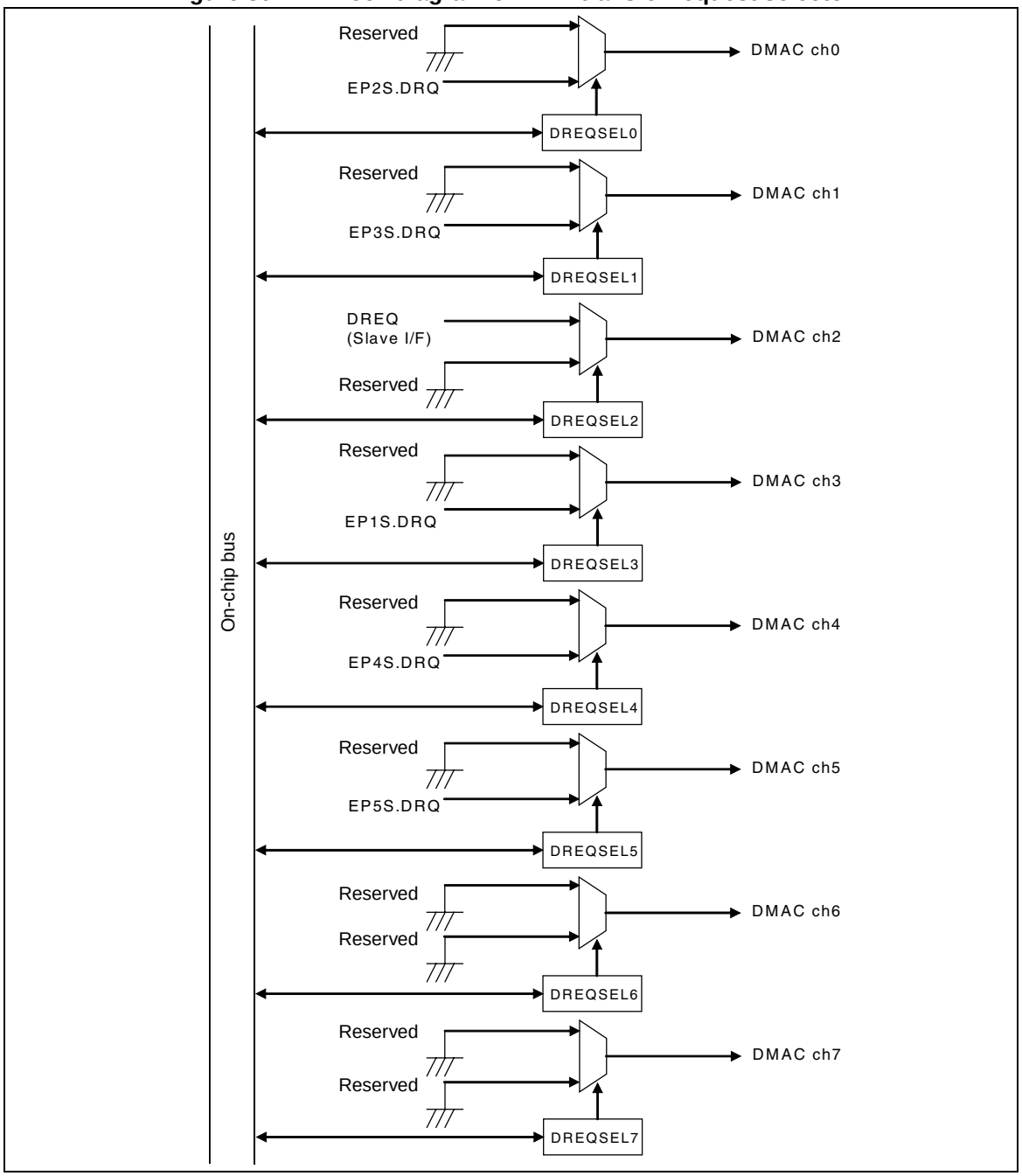
30.2 Configuration

The configuration of DMA Transfer Request is shown below:

■ Block diagram of DMA transfer request selector

Figure 30.2-1 shows the block diagram of DMA transfer diagram.

Figure 30.2-1 Block diagram of DMA transfer request selector



30.3 Registers

This section explains the configuration and functions of registers used by the DMA Transfer Request selector.

■ Registers of DMA Transfer Request selector

Table 30.3-1 lists the registers of the DMA Transfer Request selector.

Table 30.3-1 Registers of the DMA Transfer Request selector

Abbreviated Register Name	Register Name	Reference
DREQSEL	DREQ select register	30.3.1

30.3.1 DREQ Select Register (DREQSEL)

This register selects DREQ.

Figure 30.3-1 shows the bit configuration of the DREQ select register (DREQSEL).

Figure 30.3-1 Bit configuration of the DREQ select register (DREQSEL)

bit	7	6	5	4	3	2	1	0
	DREQSEL7	DREQSEL6	DREQSEL5	DREQSEL4	DREQSEL3	DREQSEL2	DREQSEL1	DREQSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	1	1	1	0	1	1

R/W: Read/Write

<Note>

Be sure to access this register in units of bytes or halfwords.

[bit7 to bit0]: DREQSEL7 to DREQSEL0 (DREQ select bits)

The bit numbers of DREQSEL7 to DREQSEL0 correspond to channel numbers of the DMA controller (DMAC). DREQ is set for each channel.

Channel	DREQ Select Bit	Written Value	Explanation
0	DREQSEL0	0	Reserved
		1	EP2S.DRQ (64 bytes) (interrupt, bulk transfer)
1	DREQSEL1	0	Reserved
		1	EP3S.DRQ (64 bytes) (interrupt, bulk transfer)
2	DREQSEL2	0	DREQ (slave interface)
		1	Reserved
3	DREQSEL3	0	Reserved
		1	EP1S.DRQ (256 bytes) (interrupt, bulk transfer)
4	DREQSEL4	0	Reserved
		1	EP4S.DRQ (64 bytes) (interrupt, bulk transfer)
5	DREQSEL5	0	Reserved
		1	EP5S.DRQ (64 bytes) (interrupt, bulk transfer)
6	DREQSEL6	0	Reserved
		1	Reserved
7	DREQSEL7	0	Reserved
		1	Reserved

<Note>

If the written value is reserved, "0" is output.

CHAPTER 31 USB Function

This chapter explains the functions and operations of the USB function.

31.1 Overview

31.2 Configuration

31.3 Registers

31.4 Explanation of Operations and Setting Procedure Examples

31.1 Overview

The USB function is the interface that supports the Universal Serial Bus (USB) communication protocol. The FULL transfer speed (12 Mbps) is supported.

■ Overview

The USB function supports the USB communication protocol. Basic protocol operations (handshake) are supported by the hardware, and only the processing of USB communication data is required for implementing USB communication.

The USB function has the following features:

- Support of the FULL speed (12 Mbps)
- Automatic response for the device status
- Automatic generation and checking for Bit Stripping, Bit Stuffing, CRC5, and CRC16
- Toggle check using a data synchronization bit
- Automatic response to all standard commands except the Get/SetDescriptor and SynchFrame commands (For the Get/SetDescriptor and SynchFrame commands, processing similar to that of class vendor commands can be executed.)
- Class vendor commands are received as data, and the firmware can respond to these commands.
- Support of up to six endpoints (Endpoint 0 is provided for the fixed use of control transfers.)
- Two transfer data buffers included in each endpoint (One of the two buffers included in endpoint 0 is dedicated for IN transfers, and the other for OUT transfers.)
- Support of automatic data transfer mode using DMA (except for the buffers in endpoint 0)

<Note>

In case of using USB function, use on-chip bus clock (HCLK) with 13 MHz or over.

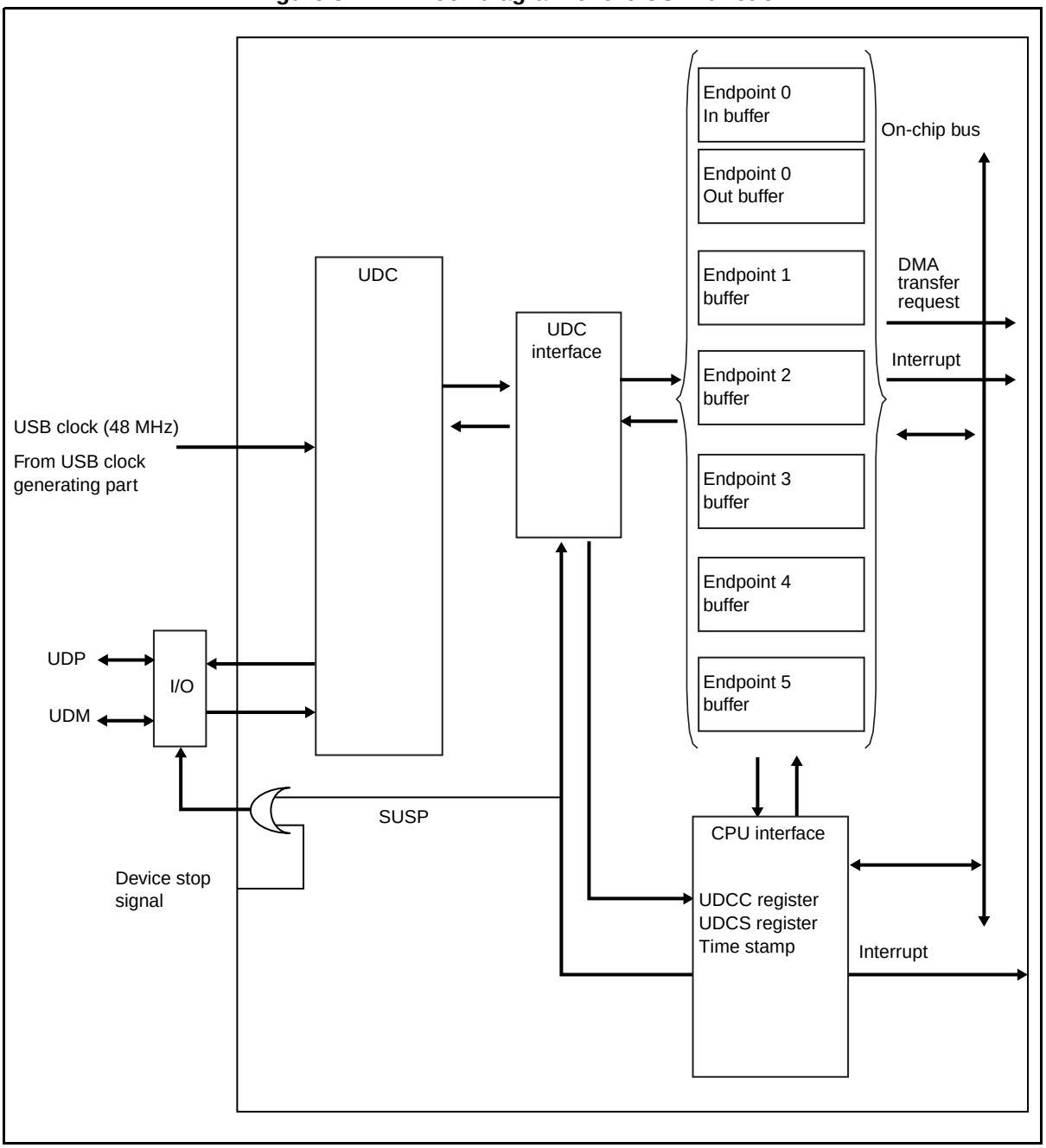
31.2 Configuration

This section shows the configuration of the USB function.

■ Block diagram of the USB function

Figure 31.2-1 is a block diagram of the USB function.

Figure 31.2-1 Block diagram of the USB function



31.3 Registers

This section explains the configuration and functions of the registers used by the USB function.

■ USB function register list

Table 31.3-1 lists the registers of the USB function.

Table 31.3-1 Registers of the USB function (1 / 2)

Abbreviated Register Name	Register Name	Reference
UCCR	USB Clock Setting Register	29.3.1
USBSEL	USB Selection Register	31.3.1
USBEN	USB Enable Register	31.3.2
UDCC	UDC control register	31.3.3
EP0C	EP0 control register	31.3.4
EP1C	EP1 control register	31.3.5
EP2C	EP2 control register	31.3.5
EP3C	EP3 control register	31.3.5
EP4C	EP4 control register	31.3.5
EP5C	EP5 control register	31.3.5
TMSP	Time stamp register	31.3.6
UDCS	UDC status register	31.3.7
UDCIE	UDC enable interrupt request register	31.3.8
EP0IS	EP0I status register	31.3.9
EP0OS	EP0O status register	31.3.10
EP1S	EP1 status register	31.3.11
EP2S	EP2 status register	31.3.11
EP3S	EP3 status register	31.3.11
EP4S	EP4 status register	31.3.11
EP5S	EP5 status register	31.3.11
EP0DTH	EP0 data register upper	31.3.12
EP0DTL	EP0 data register lower	31.3.12
EP1DTH	EP1 data register upper	31.3.12
EP1DTL	EP1 data register lower	31.3.12

Table 31.3-1 Registers of the USB function (2 / 2)

Abbreviated Register Name	Register Name	Reference
EP2DTH	EP2 data register upper	31.3.12
EP2DTL	EP2 data register lower	31.3.12
EP3DTH	EP3 data register upper	31.3.12
EP3DTL	EP3 data register lower	31.3.12
EP4DTH	EP4 data register upper	31.3.12
EP4DTL	EP4 data register lower	31.3.12
EP5DTH	EP5 data register upper	31.3.12
EP5DTL	EP5 data register lower	31.3.12

■ Register bit updating timing UDCC.RST dependency

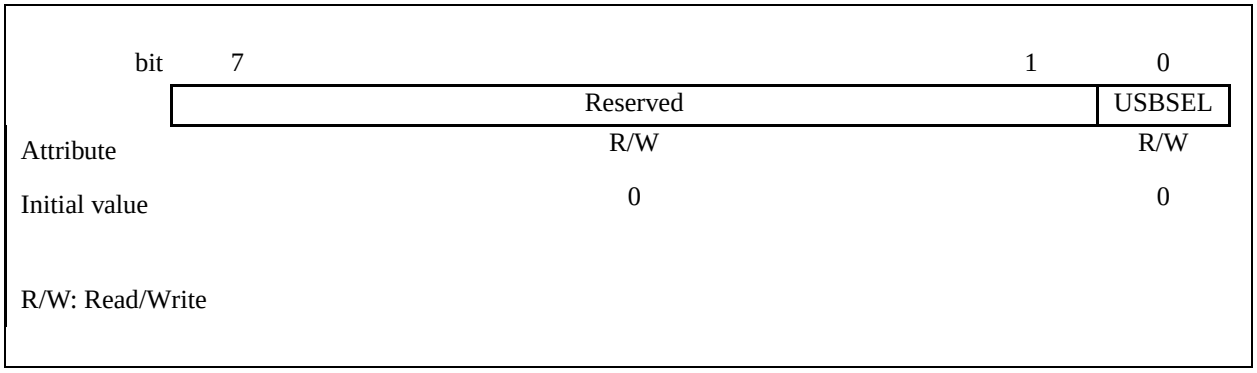
	Register	Bit
Register bit to be updated at UDCC.RST=1	UDCC	HCONX, RFBK, PWC
	EP0C	PKS0
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
	EP3C	EPEN, TYPE, DIR, PKS3
	EP4C	EPEN, TYPE, DIR, PKS4
	EP5C	EPEN, TYPE, DIR, PKS5
Register bit to be initialized at UDSS.RST=1 (Update the register bit at UDCC.RST=0)	EP0IS	BFINI, DRQI
	EP0OS	BFINI, DRQO, SPK
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
	EP3S	BFINI, DRQ, SPK
	EP4S	BFINI, DRQ, SPK
	EP5S	BFINI, DRQ, SPK
	TMSP	TMSP
	UDCS	SUSP, SOF, BRST, WKUP, SETP, CONF
	UDCIE	SUSPIE, SOFIE, BRSTIE, WKUPIE, CONFN, CONFIE
Register bit not to be affected by UDCC.RST.	UDCC	RESUME, USTP
	EP0C	STAL
	EP1C	DMAE, NULE, STAL
	EP2C	DMAE, NULE, STAL
	EP3C	DMAE, NULE, STAL
	EP4C	DMAE, NULE, STAL
	EP5C	DMAE, NULE, STAL
	EP1DTH/L	BFDT
	EP2DTH/L	BFDT
	EP3DTH/L	BFDT
	EP4DTH/L	BFDT
	EP5DTH/L	BFDT

31.3.1 USB Selection Register (USBSEL)

This register selects USB.

Figure 31.3-1 shows the bit configuration of the USB selection register.

Figure 31.3-1 Bit Configuration of USB Selection Register.



<Note>

Access this register by byte or half-word.

[bit7 to bit1]: Reserved bit

In case of writing	Write "0".
In case of reading	"0" is read.

[bit0]: USBSEL (USB Selection Bit)

This bit selects USB.

Written Value	Explanation
0	Uses USB function with Mini-Host.
1	The setting is prohibited.

31.3.2 USB Enable Register (USBEN)

This register enables the usage of USB.

Figure 31.3-2 shows the bit configuration of USB enable register.

Figure 31.3-2 Bit Configuration of USB Enable Register

bit	7	...	1	0
	Reserved			USBEN
Attribute	R/W			R/W
Initial value	0			0

R/W: Read/Write

<Note>

Access this register by byte.

[bit7 to bit1]: Reserved bit

In case of writing	Write "0".
In case of reading	"0" is read.

[bit0]: USBEN (USB Enable Bit)

Enables the use of USB function with Mini-HOST.

Written Value	Explanation
0	Disable the use of USB function with Mini-HOST.
1	Enable the use of USB function with Mini-HOST.

<Note>

Write "1" in this bit to use USB function with Mini-HOST.

31.3.3 UDC Control Register (UDCC)

This register controls the UDC core circuit.

Figure 31.3-3 shows the bit configuration of the UDC control register (UDCC).

Figure 31.3-3 Bit configuration of the UDC control register (UDCC)

bit	7	6	5	4	3	2	1	0
	RST	RESUM	HCONX	USTP	Reserved	Reserved	RFBK	PWC
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	1	0	0	0	0	0

R/W: Read/Write

[bit7]: RST (function ReSeT bit)

This bit individually resets the USB function. This reset is equivalent to the system reset of the chip. The bit is used to reset the USB function when the host is connected via a cable. The initial value of this bit is "1" and this bit can be reset. To cancel the reset condition, write "0" to this bit.

Written Value	Explanation
0	Cancels the reset condition of the USB function.
1	Resets the USB function.

<Note>

This bit initializes the relevant bits in the time stamp register (TMSP), UDC status register (UDCS), and UDC enable interrupt request register (UDCIE) at one time. Also, it initializes the BFINI bit in each of the status registers (EP0IS, EO0OS, EP1S to EP5S) at one time. After making the initial setting, clear this bit (the BFINI bits are not cleared), and then clear the BFINI bits of the endpoints used.

[bit6]: RESUM (RESUMe set bit)

If remote wakeup is enabled (the DEVICE_REMOTE_WAKEUP bit has been set by the host with the SET_FEATURE command) and the suspended state is set, writing "1" in this bit sets the USB device to start the RESUM. To release the RESUM, write "0" in this bit.

Written Value	Explanation
0	Releases the resume start instruction from the USB device.
1	Issues the RESUM start instruction from the USB device.

<Note>

Write "1" in this bit only when the USB device is in the suspended state and remote wakeup is in the enabled state.

[bit5]: HCONX (Host CONnection bit)

This bit is used so that the USB device recognizes the connection with the host or hub.

Written Value	Explanation
0	Connected with the host or hub.
1	Disconnected from the host or hub.

<Notes>

- Even when the external pull-up resistor is set to ON and the connection from the host or hub is recognized, the bus reset and command of the USB bus are ignored if this bit is "1".
- Change this bit when "RST" of bit7 in UDC Control Register (UDCC) is "1".

[bit4]: USTP (Udc SToP bit)

This bit stops the clock of the USB operation block. When no USB operation is to be performed, "1" is written to the bit to stop the clock of the USB operation block, and this can reduce power dissipation.

Written Value	Explanation
0	Sets normal mode.
1	Stops the clock of the USB operation block.

<Notes>

- Before stop mode is set, write "1" in this bit. To cancel stop mode, write "0" in the SUSP bit in the UDC status register (UDCS), and then write "0" in the USTP bit.
- When the full speed function is used, write "0" in this bit three cycles after writing "1" in the RST bit, so that this bit is definitely reset. This bit and the RST bit can be cleared at the same time.

[bit3, bit2]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit1]: RFBK (Rate Feed BacK mode bit)

This bit specifies a data toggle mode for interrupt transfers.

Written Value	Explanation
0	Selects the alternating data toggle mode. The data PID is toggled when transfer is completed successfully.
1	Selects the data toggle mode. The data PID is unconditionally toggled.

<Note>

Change this bit when RST of bit7 in CDC Control Register (UDCC) is "1".

[bit0] :PWC (PoWer Control bit)

This bit specifies an operating power mode (self-powered or bus-powered) for the USB function. This information is reflected in the GetStatus standard command.

Written Value	Explanation
0	Uses bus-powered as the operating power source.
1	Uses self-powered as the operating power source.

<Note>

Change this bit when RST of bit7 in UDC Control Register is "1".

31.3.4 EP0 Control Register (EP0C)

This register controls endpoint 0.

Figure 31.3-4 shows the bit configuration of the EP0 control register (EP0C).

Figure 31.3-4 Bit configuration of the EP0 control register (EP0C)

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Reserved	Reserved	STAL	Reserved
Attribute	-	-	-	-	-	-	R/W	-
Initial value	X	X	X	X	0	0	0	0

bit	7	6						0
	Reserved	PKS0						
Attribute	-	R/W						
Initial value	0	1000000						

R/W: Read/Write

-: Undefined

X: Undefined

[bit15 to bit12]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit11, bit10]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit9]: STAL (STALl ep0 set bit)

This bit specifies the STALL state (STALL response) for endpoint 0.

Written Value	Explanation
0	Releases the endpoint from the STALL state.
1	Sets the STALL state.

<Note>

While this bit is "1", the STALL response is continued for the host. After "0" is written to this bit, a normal SETUP packet is received and the device returns from the STALL state.

[bit8, bit7]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit6 to bit0]: PKS0 (PacKet Size ep0 set bit)

These bits specify the maximum number of transfer bytes in one packet. The maximum number of transfer bytes that can be specified in one packet for endpoint 0 is 64 bytes, which is common to IN and OUT transfers.

Writing "08_H" specifies 8 bytes to be transferred, and writing "40_H" specifies 64 bytes to be transferred.

<Notes>

- Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the EP0I/O status register (EP0I/OS) are "1". Do not change the value of these bits during a USB operation.
- Do not write a value that is "40_H" or greater or "00_H" in these bits.

31.3.5 EP1 to 5 Control Registers (EP1C to EP5C)

These registers control endpoints 1 to 5.

Figure 31.3-5 shows the bit configuration of the EP1 to 5 control registers (EP1C to EP5C).

Figure 31.3-5 Bit configuration of the EP1 to 5 control registers (EP1C to EP5C)

EP1 control register (EP1C)

bit	15	14	13	12	11	10	9	8
	EPEN	TYPE		DIR	DMAE	NULE	STAL	PSK1
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Initial value	0	11		0	0	0	0	1

bit	7	0
	PSK1	
Attribute	R/W	
Initial value	00000000	

EP2 to 5 control registers (EP2C to EP5C)

bit	15	14	13	12	11	10	9	8
	EPEN	TYPE		DIR	DMAE	NULE	STAL	Reserved
Attribute	R/W	R/W		R/W	R/W	R/W	R/W	-
Initial value	0	11		0	0	0	0	0

bit	7	6	0
	Reserved	PSK2 to PSK5	
Attribute	-	R/W	
Initial value	0	1000000	

R/W: Read/Write

-: Undefined

[bit15]: EPEN (EndPoint1 to 5 ENable bit)

This bit enables or disables the endpoint. If the endpoint is enabled, it is configured by the host as an endpoint used for the USB function, and the TYPE bit, DIR bit, and PKS bit in this register become valid in configuration information.

Written value	Explanation
0	Disables the endpoint.
1	Enables the endpoint.

<Note>

Set this bit when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of this bit during a USB operation.

[bit14, bit13]: TYPE (endpoint TYPE bit)

These bits specify a transfer type supported by the endpoint.

Written Value	Explanation
00	This specification is prohibited.
01	This specification is prohibited.
10	Specifies bulk transfer.
11	Specifies interrupt transfer.

<Note>

Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of these bits during a USB operation.

[bit12]: DIR (endpoint DIRection bit)

This bit specifies a transfer direction supported by the endpoint.

Written Value	While in USB Function Operation Mode	While in Host Operation Mode (for Either EP1 or EP2 Only)
0	Specifies an OUT endpoint.	Specifies an IN endpoint.
1	Specifies an IN endpoint.	Specifies an OUT endpoint.

<Note>

Set this bit when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of this bit during a USB operation.

[bit11]: DMAE (DMA Enable bit)

This bit sets or cancels automatic buffer transfer mode. Automatic buffer transfer mode uses DMA to write transfer data to or read transfer data from the transmission/reception buffer, and it automatically transfers as much transmission/reception data as specified by the data transfer count set for DMA, in sync with IN/OUT data requests from the host.

Written Value	Explanation
0	Cancels automatic buffer transfer mode.
1	Sets automatic buffer transfer mode.

<Note>

If "1" is written to this bit and automatic buffer transfer mode is set, do not access the transmission/reception buffers from the CPU. Also, for data transfer in the OUT direction, specify the multiples of the number specified in a PKS bit in this register as the DMA transfer count.

[bit10]: NULE (NULl Enable set bit)

This bit sets or cancels NULL automatic transfer mode. If automatic buffer transfer mode is set (DMAE = 1), NULL automatic transfer mode automatically transfers 0-byte data upon detection of the last packet transfer for a data transfer request in the IN direction.

Written Value	Explanation
0	Cancels NULL automatic transfer mode.
1	Sets NULL automatic transfer mode.

<Note>

If data transfer is in the OUT direction or automatic buffer transfer mode is not set, the value of this bit does not affect communication.

[bit9]: STAL (STALl set bit)

This bit sets the STALL state (STALL response) for the endpoint or releases it from the STALL state.

Written Value	Explanation
0	Releases the endpoint from the STALL state.
1	Sets the STALL state.

<Note>

While this bit is "1", the STALL response is continued for the host. When the ClearFeature command is issued from the host after "0" is written to this bit, the device returns from the STALL state.

[bit8, bit7]: EP2 to EP5 reserved bits

These bits in the EP2 to 5 control registers (EP2C to EP5C) are reserved.

In case of writing	Write "0".
In case of reading	"0" is read.

[bit8 to bit0]: PKS1 (PacKet Size ep1 set bits)

[bit6 to bit0]: PKS2 to PKS5 (PacKet Size ep2 to ep5 set bits)

These bits specify the maximum transfer count in one packet. The following table lists the maximum transfer count in a single packet for each of endpoints 1 to 5:

Endpoint	Maximum Transfer Count	Allowable Setting Range
1	256 bytes (an odd number can also be specified)	001_H to 100_H
2 to 5	64 bytes (an odd number can also be specified)	01_H to 40_H

<Notes>

- Set these bits when both the RST bit in the UDC control register (UDCC) and the BFINI bit in the relevant register among the EP0 to 5 status registers (EP1S to EP5S) are "1". Do not change the value of these bits during a USB operation.
- Do not write a value that is equal to or greater than the maximum transfer count ("100_H" or "40_H") or "00" in these bits.
- Set "00" in bit8 and bit7 for endpoints 2 to 5.
- When using automatic buffer transfer mode (DMAE = 1), do not set a value of "0_H" to "2_H" to the relevant endpoint.

31.3.6 Time Stamp Register (TMSP)

This register indicates the frame number as of the SOF packet receipt time.

Figure 31.3-6 shows the bit configuration of the time stamp register (TMSP).

Figure 31.3-6 Bit configuration of time stamp register (TMSP)

bit	15	14	13	12	11	10	8
	Undefined	Undefined	Undefined	Undefined	Undefined	TMSP	
Attribute	-	-	-	-	-	R	
Initial value	X	X	X	X	X	000	

bit	7	0
	TMSP	
Attribute	R	
Initial value	00000000	

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit15 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10 to bit0]: TMSP (TiMe StamP bits)

These bits indicate the frame number as of the SOF packet reception time. The frame number is updated when an SOF packet is received.

31.3.7 UDC Status Register (UDCS)

This register indicates the bus status concerning USB communication and the command receipt status for a particular command. The bits except the SETP bit are interrupt resources. An interrupt request to the CPU is generated when a corresponding interrupt enable bit is enabled.

Figure 31.3-7 shows the bit configuration of the UDC status register (UDCS).

Figure 31.3-7 Bit configuration of the UDC status register (UDCS)

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	SUSP	SOF	BRST	WKUP	SETP	CONF
Attribute	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	X	X	0	0	0	0	0	0

R/W: Read/Write

-: Undefined

X: Undefined

[bit7, bit6]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit5]: SUSP (SUSPend bit)

This bit indicates whether the USB function has transferred to the suspended state. The bit is an interrupt resource.

SUSP	In Case of Reading	In Case of Writing
0	The suspended state has not been detected.	The interrupt resource is cleared.
1	The USB function has transferred to the suspended state.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit4]: SOF (Start Of Frame bit)

This bit indicates whether an SOF packet has been received and the value of the time stamp register (TMSP) has been updated. The bit is an interrupt resource.

SOF	In Case of Reading	In Case of Writing
0	An SOF packet has not been received.	The interrupt resource is cleared.
1	An SOF packet has been received.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit3]: BRST (Bus ReSeT bit)

This bit indicates whether a USB bus reset has been detected. The bit is an interrupt resource.

BRST	In Case of Reading	In Case of Writing
0	No USB bus reset has been detected.	The interrupt resource is cleared.
1	A USB bus reset has been detected.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- When a USB bus reset is detected with this bit, use the RST bit in the UDC control register (UDCC) to initialize the USB function, and make register settings again.

[bit2]: WKUP (WaKe UP bit)

This bit indicates whether the USB function has returned from the suspended state. The two resources for the return are remote wakeup because "1" is written to the RESUM bit in the UDC control register (UDCC) and wakeup because of a request from the host. This bit is automatically set when the host requests the return. The bit is an interrupt resource.

WKUP	In Case of Reading	In Case of Writing
0	No return request (resume) by the host has been detected.	The interrupt resource is cleared.
1	A return request (resume) by the host has been detected.	Ignored

---

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- Even when a return request from the host is detected, this bit is not set if "1" is set in the RESUM bit in the UDC control register (UDCC).

[bit1]: SETP (SETuP bit)

This bit indicates whether data has been received for the setup stage of a USB control transfer.

SETP	In Case of Reading	In Case of Writing
0	No data has been received for the setup stage of a control transfer.	The resource is cleared.
1	Data has been received for the setup stage of a control transfer.	Ignored

---

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- This bit is not set for an automatic response for a standard command. The bit is not an interrupt resource.

[bit0]: CONF (CONFiguration bit)

This bit indicates whether USB function configuration has been completed. The bit is set when the SetConfig USB command has been received correctly. The bit is an interrupt resource.

CONF	In Case of Reading	In Case of Writing
0	SetConfig has not been detected.	The interrupt resource is cleared.
1	SetConfig has been detected.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

31.3.8 UDC Enable Interrupt Request Register (UDCIE)

This register enables interrupts by the interrupt resource of the UDC status register (UDCS) corresponding to a specific bit (except the CONFN bit).

Figure 31.3-8 shows the bit configuration of the UDC enable interrupt request register (UDCIE).

Figure 31.3-8 Bit configuration of UDC enable interrupt request register (UDCIE)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	SUSPIE	SOFIE	BRSTIE	WKUPIE	CONFN	CONFIE
Attribute	-	-	R/W	R/W	R/W	R/W	R	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

-: Undefined

[bit7, bit6]: Reserved bits

In case of writing	Write "0".
In case of reading	"0" is read.

[bit5]: SUSPIE (SUSP Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the SUSP bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the SUSP resource.
1	Enables interrupts by the SUSP resource.

[bit4]: SOFIE (SOF Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the SOF bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the SOF resource.
1	Enables interrupts by the SOF resource.

[bit3]: BRSTIE (BRST Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the BRST bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the BRST resource.
1	Enables interrupts by the BRST resource.

[bit2]: WKUPIE (WKUP Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the WKUP bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the WKUP resource.
1	Enables interrupts by the WKUP resource.

[bit1]: CONFN (CONFigration Number bit)

This bit indicates the configuration number. The bit is updated when the interrupt resource of the CONF bit in the UDC status register (UDCS) is set.

Read Value	Explanation
0	The configuration number is 0.
1	The configuration number is 1.

[bit0]: CONFIE (CONF Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the CONF bit in the UDC status register (UDCS).

Written Value	Explanation
0	Disables interrupts by the CONF resource.
1	Enables interrupts by the CONF resource.

31.3.9 EP0I Status Register (EP0IS)

This register indicates the status regarding IN-direction transfers of endpoint 0.

Figure 31.3-9 shows the bit configuration of the EP0I status register (EP0IS).

Figure 31.3-9 Bit configuration of the EP0I status register (EP0IS)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIIE	Undefined	Undefined	Undefined	DRQI	Undefined	Undefined
Attribute	R/W	R/W	-	-	-	R/W	-	-
Initial value	1	0	X	X	X	1	X	X

bit	7	6	5	4	3	2	1	0
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Attribute	-	-	-	-	-	-	-	-
Initial value	X	X	X	X	X	X	X	X

R/W: Read/Write

-: Undefined

X: Undefined

[bit15]: BFINI (BuFfer INItial bit)

This bit initializes the transfer data transmit buffer. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the transmit buffer.
1	Initializes the transmit buffer.

<Note>

The transmit buffer and the DRQI bit are initialized when "1" is written to this bit. Before initializing the transmit buffer, first verify that "1" is written to the DRQI bit or DRQO bit in the EP0O status register (EP0OS) and there is no access from the host, and then write "1" in the STAL bit in the EP0 control register (EP0C) as necessary.

[bit14]: DRQIIE (Data ReQuest In Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the DRQI bit.

Written Value	Explanation
0	Disables interrupts by the DRQI resource.
1	Enables interrupts by the DRQI resource.

[bit13 to bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10]: DRQI (Data ReQuest In bit)

This bit indicates whether an IN packet transfer from the host for endpoint 0 has been completed correctly and the data has been read from the transmit buffer; in other words, it indicates whether the next transmit data can be written to the buffer. The bit is an interrupt resource.

DRQI	In Case of Reading	In Case of Writing
0	Transmit data cannot be written.	The interrupt resource is cleared.
1	Transmit data can be written.	Ignored

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit9 to bit0]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

31.3.10 EP0O Status Register (EP0OS)

This register indicates the status regarding OUT-direction transfers of endpoint 0.

Figure 31.3-10 shows the bit configuration of the EP0O status register (EP0OS).

Figure 31.3-10 Bit configuration of EP0O status register (EP0OS)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQOIE	SPKIE	Undefined	Undefined	DRQO	SPK	Reserved
Attribute	R/W	R/W	R/W	-	-	R/W	R/W	-
Initial value	1	0	0	X	X	0	0	0

bit	7	6						0
	Reserved	SIZE						
Attribute	-	R						
Initial value	0	X						

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit15]: BFINI (BuFfer INItial bit)

This bit initializes the transfer data receive buffer. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the receive buffer.
1	Initializes the receive buffer.

<Note>

The receive buffer, DRQO bit, and SPK bit are initialized when "1" is written to this bit. Before initializing the reception buffer, first verify that "1" is written to the DRQO bit or DRQI bit in the EP0I status register (EP0IS) and there is no access from the host, and then write "1" in the STAL bit in the EP0 control register (EP0C) as necessary.

[bit14]: DRQOIE (Data ReQuest Out Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the DRQO bit.

Written Value	Explanation
0	Disables interrupts by the DRQO resource.
1	Enables interrupts by the DRQO resource.

[bit13]: SPKIE (SPK Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the SPK bit.

Written Value	Explanation
0	Disables interrupts by the SPK resource.
1	Enables interrupts by the SPK resource.

[bit12, bit11]: Undefined bits

In case of writing	Ignored
In case of reading	A value is undefined.

[bit10]: DRQO (Data ReQuest Out bit)

This bit indicates whether an OUT packet transfer from the host for endpoint 0 has been completed correctly and the data has been written to the reception buffer; in other words, it indicates whether the received data can be read from the buffer. The bit is an interrupt resource.

DRQO	In Case of Reading	In Case of Writing
0	Received data cannot be read.	The interrupt resource is cleared.
1	Received data can be read.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- Clear this bit by writing "0" in the bit after reading data from the reception buffer. However, write "0" in the bit only if "1" has been written to it.
- If "1" is held in this bit, the reception buffer is not updated. The buffer can be updated while "0" is written to the bit. If the bit is "1" when an OUT packet is requested, the NAK response is automatically returned to the host.

[bit9]: SPK (Short PacKet bit)

This bit indicates that packet (including 0-byte packets) of the transfer data received correctly from the host has not reached the maximum packet count specified in the PKS bit in the EP0 control register (EP0C). The bit is an interrupt resource.

SPK	In Case of Reading	In Case of Writing
0	Data of the max. packet count have been received.	The interrupt resource is cleared.
1	Data of less than the max. packet count have been received.	Ignored

---

<Note>

When a read-modify-write instruction is used, "1" is read.

[bit8, bit7]: Reserved bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit6 to bit0]: SIZE (packet SIZE bits)

These bits indicate the number of data bytes written to the reception buffer after the completion of an OUT packet transfer of endpoint 0. The bits are updated to valid values when the interrupt resource of the DRQO bit in the EP0O status register (EP0OS) is set. For example, when 8 bytes of data are written, these bits indicate "08_H", and when 64 bytes of data are written, these bits indicate "40_H".

31.3.11 EP1 to 5 Status Registers (EP1S to EP5S)

These registers indicate the status regarding endpoints 1 to 5.

Figure 31.3-11 shows the bit configuration of the EP1 to 5 status registers (EP1S to EP5S).

Figure 31.3-11 Bit configuration of EP1 to 5 status registers (EP1S to EP5S)

EP1 status register (EP1S)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	SIZE1
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	R
Initial value	1	0	0	X	0	0	0	X

bit	7							0
	SIZE1							
Attribute	R							
Initial value	X							

EP2 to 5 status registers (EP2S to EP5S)

bit	15	14	13	12	11	10	9	8
	BFINI	DRQIE	SPKIE	Reserved	BUSY	DRQ	SPK	Reserved
Attribute	R/W	R/W	R/W	-	R	R/W	R/W	-
Initial value	1	0	0	X	0	0	0	0

bit	7	6						0
	Reserved	SIZE 2 to SIZE 5						
Attribute	-	R						
Initial value	0	X						

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

[bit15]: BFINI (Endpoint1 to 5 ENable bit)

This bit initializes the transfer data transmit/receive buffers. The bit automatically becomes "1" when "1" is written to the RST bit in the UDC control register (UDCC). If the RST bit is "1", write "0" in the BFINI bit only after writing "0" in the RST bit.

Written Value	Explanation
0	Cancels initialization of the transmission/reception buffers.
1	Initializes the transmission/reception buffers.

<Note>

The transmission/reception buffers of endpoints 1 to 5 are configured as double buffers. Writing "1" in this bit initializes the double buffers, the DRQ bit, and the SPK bit at the same time. Before initializing the transmission/reception buffers, first verify that the DRQ bit has become "1", the BUSY bit has become "0", and there is no access from the host, and then write "1" in the STAL bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) as necessary.

[bit14]: DRQIE (Data ReQuest Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the DRQ bit.

Written Value	Explanation
0	Disables interrupts by the DRQ resource.
1	Enables interrupts by the DRQ resource.

<Note>

When using automatic buffer transfer mode, make the setting of DMA to enable the transfer, and then write "1" in the DRQIE bit.

[bit13]: SPKIE (SPK Interrupt Enable bit)

This bit enables interrupts by the interrupt resource of the SPK bit.

Written Value	Explanation
0	Disables interrupts by the SPK resource.
1	Enables interrupts by the SPK resource.

[bit12]: Reserved bit

In case of writing	Ignored
In case of reading	A value is undefined.

[bit11]: BUSY (BUSY flag bit)

This bit indicates that the host is writing data to or reading data from the transmission/reception buffer. The bit is automatically updated.

Read Value	Explanation
0	There is no access from the host.
1	The host is writing or reading data.

<Note>

If "1" is indicated by this bit and the DRQ bit, a buffer different from the one accessed by the CPU or DMA is being accessed by the host. Usually, the control with the BUSY bit is not required. Before initializing the transmission/reception buffer by using the BFINI bit, verify that the DRQ bit is "1", the BUSY bit is "0", and there is no access from the host, and then write "1" in the STAL bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C).

[bit10]: DRQ (Data ReQuest bit)

This bit indicates that a packet transfer of one of endpoints 1 to 5 has been completed correctly, and data processing is required. The bit is an interrupt resource.

DRQ	In Case of Reading	In Case of Writing
0	The data cannot be written/read from the transmission/reception buffer.	The interrupt resource is cleared.
1	A packet transfer has been completed correctly.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- If automatic buffer transfer mode is not used, write "0" in this bit to clear it after data is written to or read from the transmission/reception buffer. The buffer being accessed is switched when the bit is cleared. After DRQ bit is cleared, DRQ of "0" might not be read. If IN is set for the transfer direction, this bit is "1", and the bit is cleared without the data in the buffer being written, data with a length of 0 bytes will be set as a result. If "1" is written to the DIR bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) in the initial settings, the DRQ bit of the relevant endpoint also becomes "1" at the same time. Also, if this bit is "0", do not write "0" in it.

[bit9]: SPK (Short PacKet bit)

This bit indicates that the transfer data from the host has been received normally but the number of transfer bytes in individual packets (including 0-byte packets) is less than the max. packet count specified per packet in the PKS bit in the relevant register among EP1 to 5 control registers (EP1C to EP5C). The bit is an interrupt resource.

SPK	In Case of Reading	In Case of Writing
0	The data of the max. packet count have been received.	The interrupt resource is cleared.
1	The data less than the max. packet count have been received.	Ignored

<Notes>

- When a read-modify-write instruction is used, "1" is read.
- If data transfer is in the IN direction, this bit does not change.

[bit8, bit7]: EP2 to EP5 reserved bits

These bits in the EP2 to 5 status registers (EP2S to EP5S) are reserved bits.

In case of writing	Ignored
In case of reading	"0" is read.

[bit8 to bit0]: SIZE1 (packet SIZE bits)

[bit6 to bit0]: SIZE2 to SIZE 5 (packet SIZE bits)

These bits indicate the number of data bytes written to the reception buffer after the completion of an OUT packet transfer of one of endpoints 1 to 5. The bits are updated to valid values when the relevant DRQ interrupt resource is generated.

The following table lists the maximum transfer count of each of endpoints 1 to 5:

Endpoint	Maximum Transfer Count	Indication Range
1	256 bytes	000_H to 100_H
2 to 5	64 bytes	00_H to 40_H

<Note>

These bits indicate the number of data bytes written to the buffer by the host for an OUT-direction transfer. The read value has no meaning for an IN-direction transfer.

31.3.12 EP0 to 5 Data Registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

These registers are used for read access or write access of the transmit/receive buffers holding transfer data regarding endpoints 0 to 5.

Figure 31.3-12 shows the bit configuration of the EP0 to 5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL).

Figure 31.3-12 Bit configuration of the EP0 to 5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL)

EP0 to 5 data registers upper (EP0DTH to EP5DTH)

bit	15							8
	BFDT							
Attribute	R/W							
Initial value	X							

EP0 to 5 data registers lower (EP0DTL to EP5DTL)

bit	7							0
	BFDT							
Attribute	R/W							
Initial value	X							

R/W: Read/Write

X: Undefined

[bit15 to bit0]: BFDT (BuFfer DaTa bit)

These registers are used to read data from or write data to the transmit/receive buffer of each endpoint.

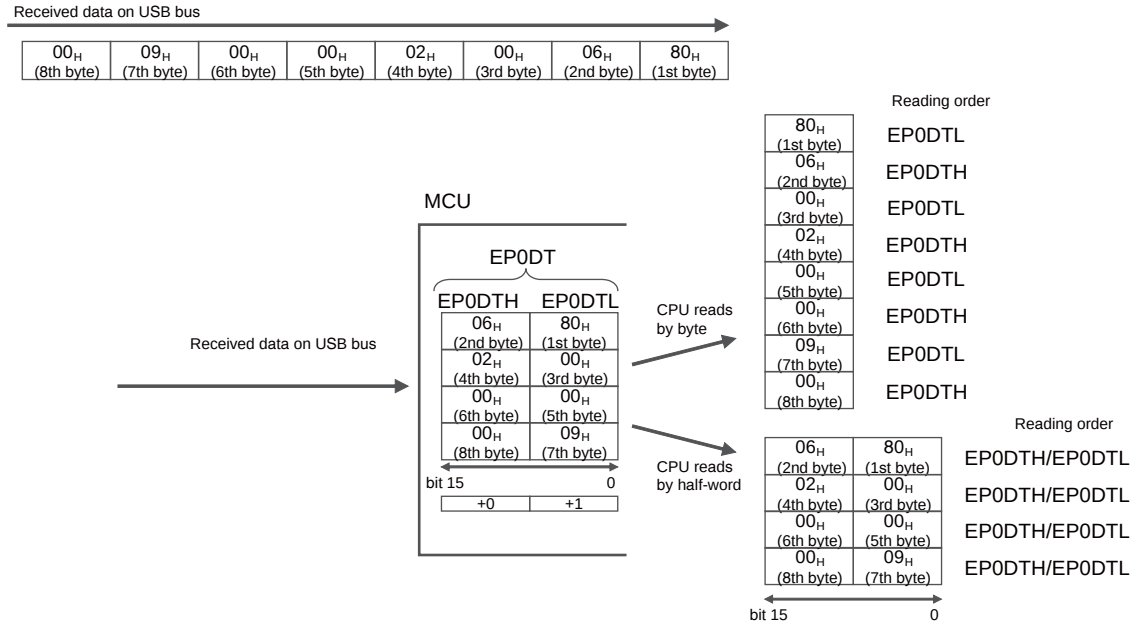
<Note>

EP0 to EP5 data registers(EP0DTH to EP5DTH/EP0DTL to EP5DTL) are little endian. The order for storing data to these registers is from the lower data (EPxDTL) to the upper data (EPxDTH). (x=1 to 5)

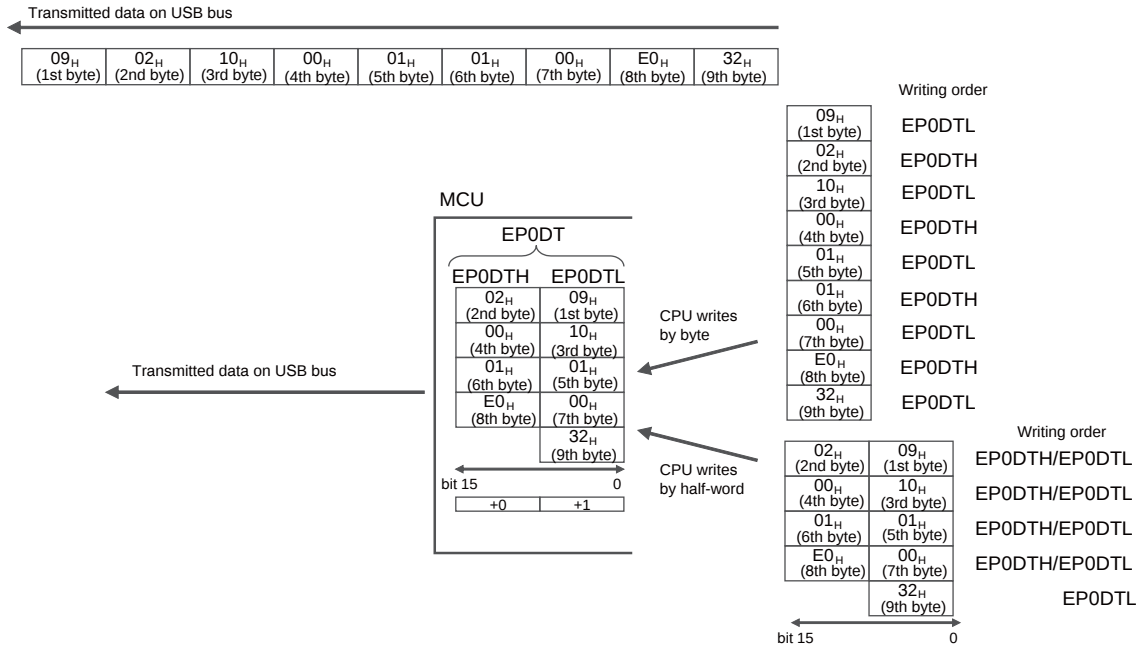
The CPU access to EP0 to EP5 data registers can be implemented by byte and half-word and by byte, the lower data (EPxDTL) is accessed first and then the upper data (EPxDTH) is accessed. The access is continued for the lower data and the upper data alternately.

For half-word access, execute the endian transformation for the transferred data. The access to these registers with bit operation instruction is prohibited.

Example) When transferring SETUP of get descriptor



Example) When transferring IN of get descriptor



The CPU access to EP0 to EP5 data registers (EP0DTH to EP5DTH/EP0DTL to EP5DTL) are implemented only by half-word access. Execute the endian transformation for the transferred data (see "Automatic buffer transfer mode" in 31.4.6 DMA Transfer Function).

31.4 Explanation of Operations and Setting Procedure Examples

This section explains operations of the USB function and also shows examples of procedures for setting the operating state are shown.

■ Overview

The USB function performs bidirectional packet transfer between a host controller that supports the USB communication protocol. The host and USB devices are connected and configured using enumeration, and communicate by using device drivers for different types of transfer.

<Note>

To use the USB function, set the USB clock generation block to enable USB clock output while USB operations are disabled by a USB enable register (USBEN) setting (USBEN = 0), and then enable USB operations (USBEN = 1).

This section uses enumeration in an example to explain operations of the USB communication between the host and a USB device, and then shows operations of registers and USB packets to provide an understanding of all the processing.

● Enumeration processing

Enumeration processing establishes the initial connection between host and the USB device during a USB operation. The host uses a USB control transfer (a type of USB transfer) to check the types of devices that have a USB connection (the procedure is specified in the USB specifications). Endpoint 0 among the six endpoints is used for this purpose (USB specification).

To use endpoint 1 to endpoint 5 for the operation, receive the following operations on USB bus:

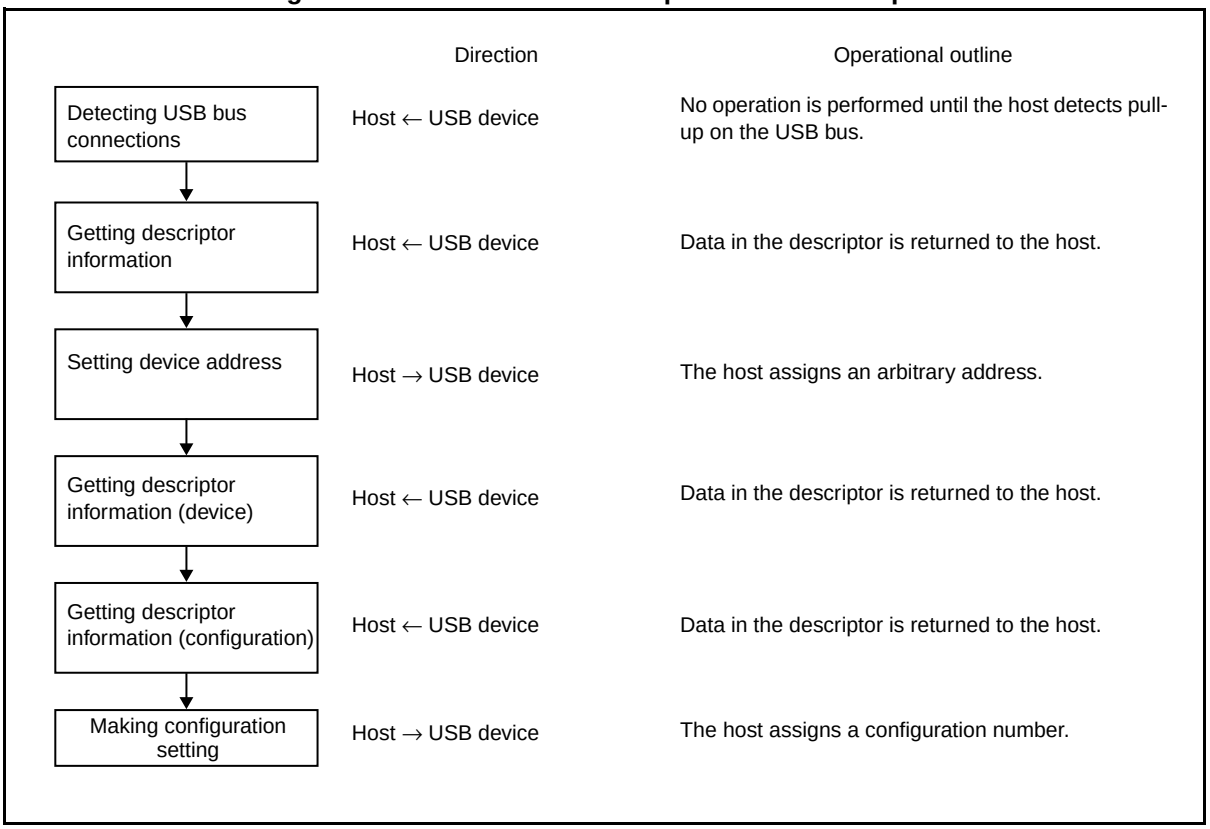
(1) USB bus reset

(2) Address setting by SET_Address

(3) Configuration setting by SET_Config

Figure 31.4-1 shows a connection example for USB cable pins.

Figure 31.4-1 Connection example for USB cable pins


Direction
Operational outline
Detecting USB bus connections
Host ← USB device
No operation is performed until the host detects pull-up on the USB bus.
Getting descriptor information
Host ← USB device
Data in the descriptor is returned to the host.
Setting device address
Host → USB device
The host assigns an arbitrary address.
Getting descriptor information (device)
Host ← USB device
Data in the descriptor is returned to the host.
Getting descriptor information (configuration)
Host ← USB device
Data in the descriptor is returned to the host.
Making configuration setting
Host → USB device
The host assigns a configuration number.

- Detecting connections

  The USB device notifies the host.

  The host monitors the two signal lines on the USB bus (D+, D-). If any of the signals is at the "H" level, the host recognizes that a device is connected.

  For details of the procedure for using a self-powered USB device, see "31.4.1 Detection of Connections and Disconnections". To use a bus-powered USB device, follow the instructions explained in "Making initial settings for registers and starting operation" below.

- Making initial settings for registers and starting operation

  The following procedure is an example for making initial settings for USB function registers.

  1. Make settings (packet size, etc.) for endpoint 0 by using the EP0 control register (EP0C).
  2. Set the EPEN bits, DIR bits, TYPE bits, etc. of endpoints 1 to 5 by using the EP1 to 5 control registers (EP1C to EP5C).
  3. Clear the RST bit in the UDC control register (UDCC).
  4. Clear the BFINI bits in the EP0I status register (EP0IS), EP0O status register (EP0OS), and EP1 to 5 status registers (EP1S to EP5S).
  5. Clear the HCONX bit in the UDC control register (UDCC).

- Resetting the USB bus

  The USB device core is initialized when the host generates the bus reset condition and the USB device is reset.

  For the USB device, perform the following operations in the order shown (this processing is not required for resetting the bus first after a USB connection is established):

  1. Write "1" in the RST bit in the UDC control register (UDCC) to initialize the USB function.
  2. Set the used transmit/receive buffers and the relevant control registers again.
  3. Restore control of the firmware to the point immediately prior to the start of enumeration.

- Getting descriptor information

  When the host sends a request to the USB device, it sends data to the host. The communication takes place over three stages: the setup stage, the data stage, and the status stage.

  - Setup stage

    In this stage, packets are checked to verify that they were received normally from the host, and the received information is decoded to determine what the command is. Also, the descriptor information to be returned in the next data stage is prepared in the transmit buffer.

  - Data stage

    Data is checked to verify that the host sent it normally.

  - Status stage

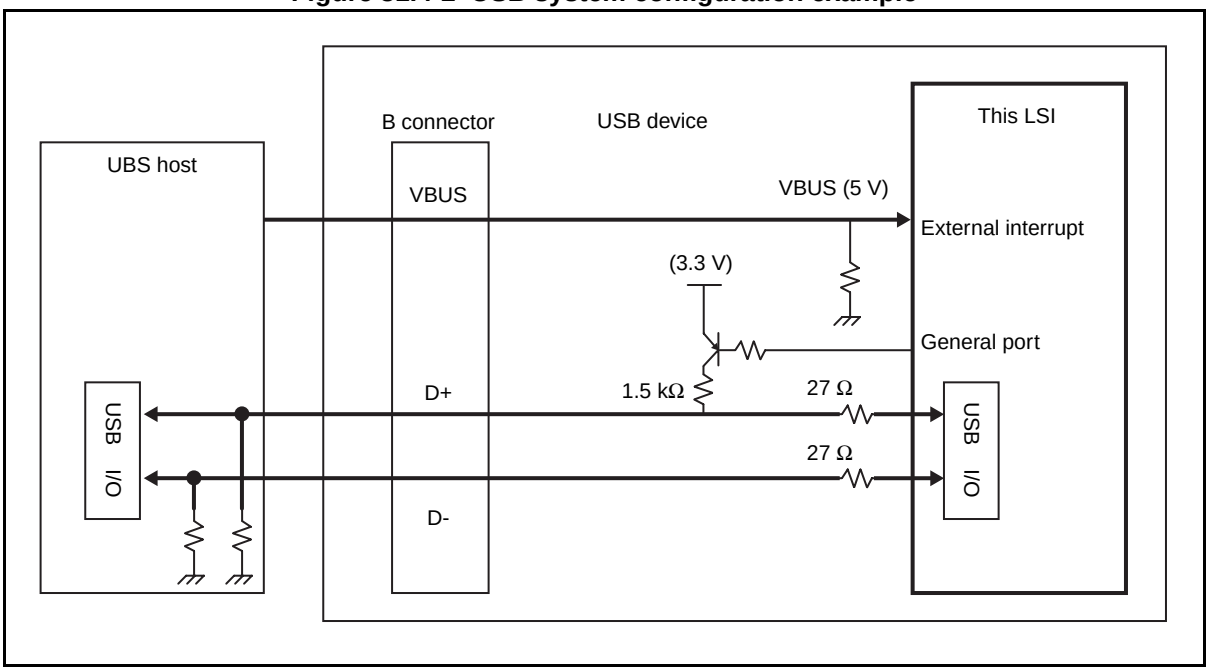
    The host transfers a packet containing no data, and termination processing is executed.

31.4.1 Detection of Connections and Disconnections

This section explains the detection of USB host connections and disconnections.

By connecting the external interrupt pin to the VBUS pin of the USB connector and connecting a pull-down resistor to it, a disconnection from the USB host can be detected. Figure 31.4-2 shows an example of connections to the D+, D-, and VBUS pins of the USB connector.

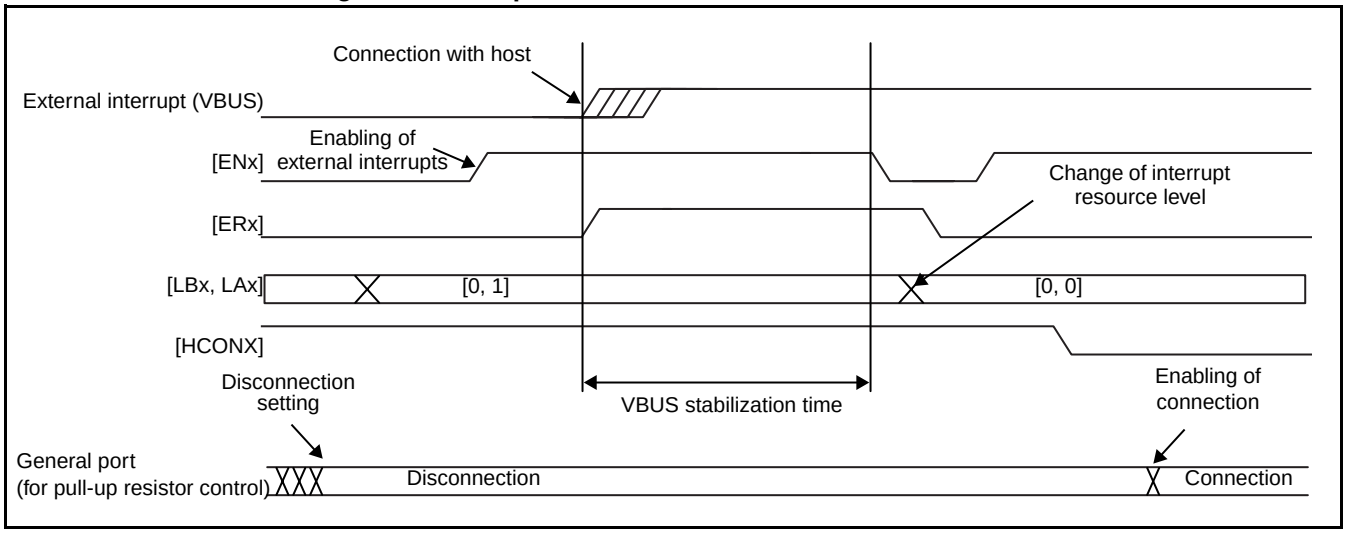
Figure 31.4-2 USB system configuration example



■ Connection detection operation

Figure 31.4-3 shows the operation at the connection detection time.

Figure 31.4-3 Operation at the connection detection time



The USB device recognizes and processes the connection to the host in the following sequence:

1. The general port for pull-up resistor control is set on the pull-up resistor disconnection side.
2. The level of the external interrupt resource connected to VBUS is set to "H" level detection, and interrupts are enabled.
3. The connection of the USB host in "H" level detection at the external interrupt pin is recognized, which is followed by a wait for the VBUS stabilization time.
4. External interrupts are disabled once.

   The external interrupt resource level is changed to "L" level detection, and the interrupt resource is cleared to enable external interrupts again.
5. Initial settings are made (all initialization is performed including that for the USB function registers).

   For details, see "Making initial settings for registers and starting operation" in this "Overview" section.
6. After the HCONX bit in the UDC control register (UDCC) is cleared, the general port pull-up resistor control is set on the pull-up resistor connection side to connect the pull-up resistor of D+ is connected.

   Even if the pull-up resistor is not controlled, clear the HCONX bit.

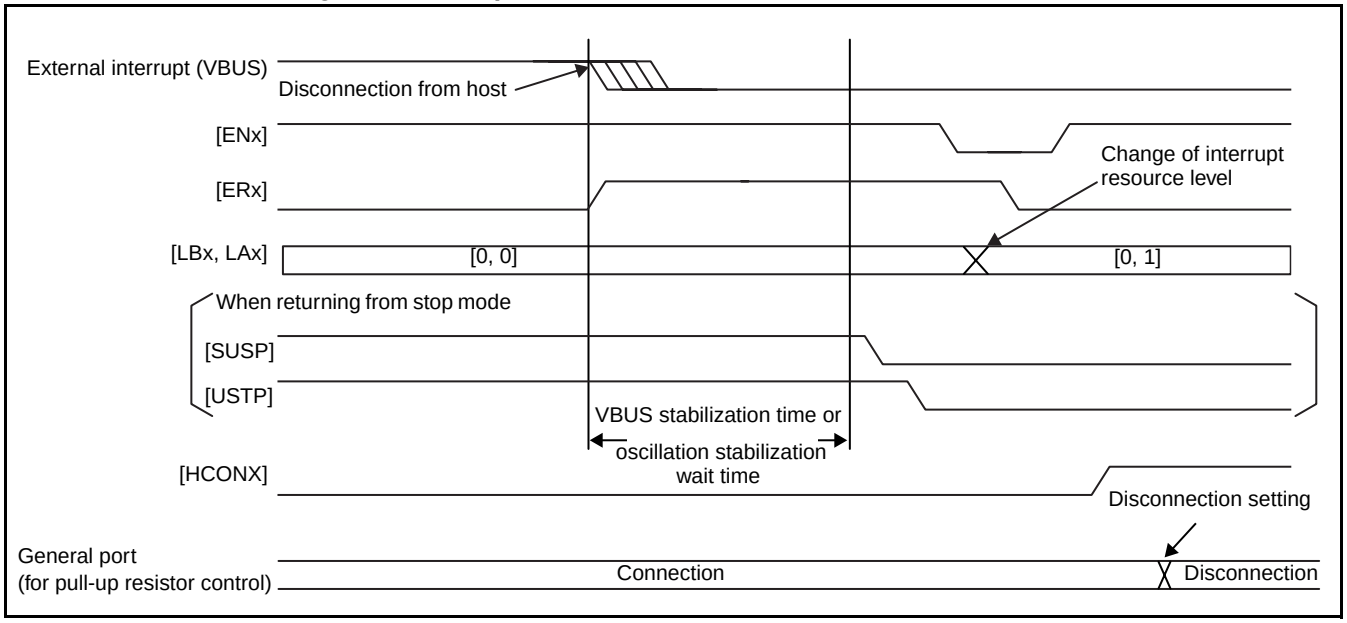
<Note>

If an external noise filter is connected to the external interrupt pin, the program need not wait for the above VBUS stabilization time.

■ Disconnection detection operation

Figure 31.4-4 shows the operation at the disconnection detection time.

Figure 31.4-4 Operation at the disconnection detection time



The USB device recognizes and processes the disconnection from the host in the following sequence:

1. The "L" level of the external interrupt pin connected to VBUS is detected, and the USB host is recognized as disconnected.

2. After the return from stop mode and the wait for the oscillation stabilization wait time, the SUSP bit in the UDC status register (UDCS) and the USTP bit in the UDC control register (UDCC) are cleared in this order.

   In any mode other than stop mode, there is a wait for the VBUS stabilization time.

3. External interrupts are disabled once.

   The external interrupt resource level is changed to the "H" level detection, and the interrupt resource is cleared to enable external interrupts again.

4. After "1" is written to the HCONX bit in the UDC control register (UDCC), the general port for pull-up resistor control is set on the pull-up resistor disconnection side to disconnect the pull-up resistor of D+.

   Even if the pull-up resistor is not controlled, write "1" in the HCONX bit.

<Note>

If an external noise filter is connected to the external interrupt pin, the program need not wait for the above VBUS stabilization time.

31.4.2 Register Operation for a Command Response

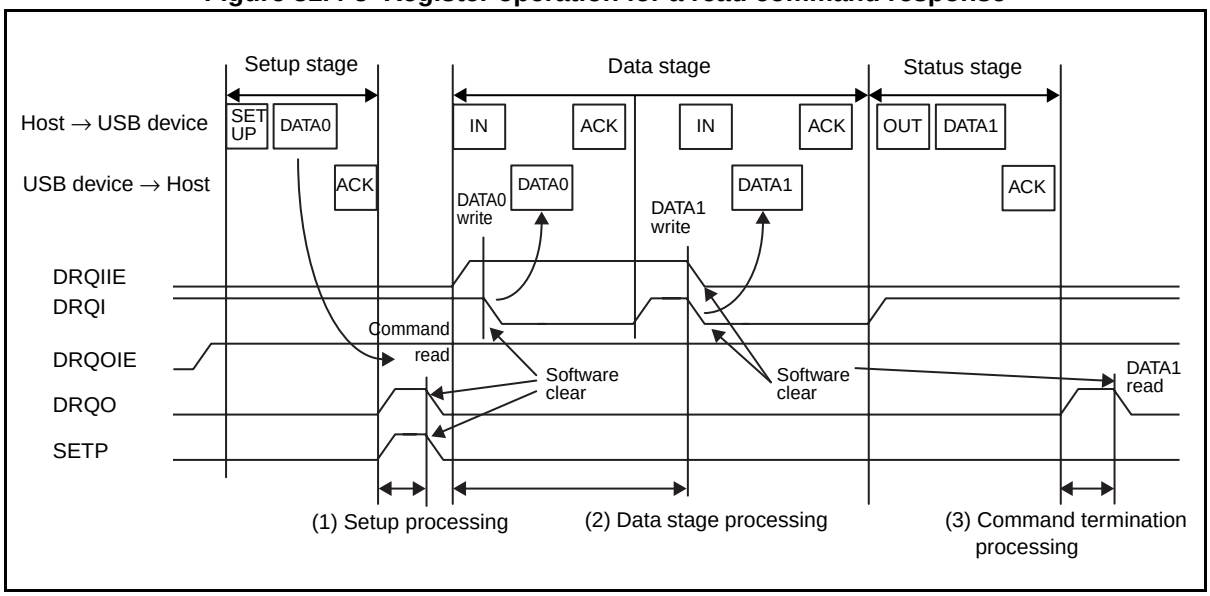
This section explains how to process USB packets through the operation and control of basic registers (architecture).

Firmware processing for a CPU interrupt is executed for each handshake. This is equivalent in meaning to executing the processing in each stage of individual packets.

■ Register operation for a read command response

Figure 31.4-5 shows the register operation for a read command response (in the case of the GetDescripter or SynchFrame command or any of the class vendor commands).

Figure 31.4-5 Register operation for a read command response



- Setup processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" when data is received for the setup stage. CPU interrupt processing begins when the DRQO bit becomes "1". The SETP bit in the UDC status register (UDCS) is verified. If the SETP bit is "1", as many commands as necessary in the receive buffer are read (it is not always necessary to read all 8 bytes) and decoded, various setup processing is executed, the SETP bit and DRQO interrupt resource bit are cleared, and then the processing returns.

- Data stage post-processing

  If the command decoding results indicate that the data stage is in the IN direction, "1" is written to the DRQIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S) to enable interrupts (which is the initial value of the DRQI bit in the EP0I status register (EP0IS), which is an interrupt resource, so "1" is just written to the DRQIE bit). Then, a CPU interrupt is used to transfer transmit data to the transmit buffer. The DRQI interrupt resource bit is cleared after the transfer is completed, and then the processing returns.

  The DRQI bit becomes "1" at the completion of the IN-direction data packet processing. CPU interrupt processing is entered when the DRQI bit becomes "1", and transmit data is transferred to the transmit buffer in preparation for the next data packet. The DRQI interrupt resource bit is cleared after the transfer is completed, and then the processing returns.

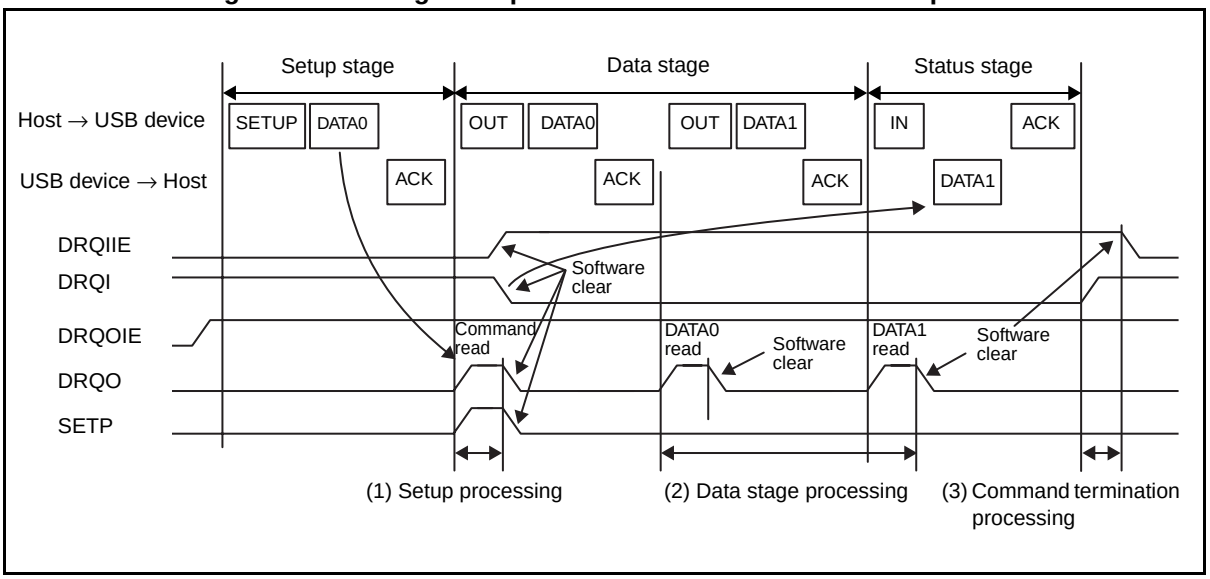
- Command termination processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" upon completion of the OUT-direction status stage. CPU interrupt processing is entered when the DRQO bit becomes "1", then the received data count is confirmed to be 0, the DRQO interrupt resource bit is cleared in preparation for the next setup stage, and then the processing returns.

■ Register operation for a write command response

Figure 31.4-6 shows the register operation for a write command response (in the case of the SetDescripter command or any of the class vendor commands).

Figure 31.4-6 Register operation for a write command response



- Setup processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" when data is received for the setup stage. CPU interrupt processing begins when the DRQO bit becomes "1". The SETP bit in the UDC status register (UDCS) is verified. If the SETP bit is "1", as many commands as necessary in the receive buffer are read (it is not always necessary to read all 8 bytes) and decoded, and then various setup processing is executed.

  In preparation for a 0-byte response in the status stage, without writing any data in the transmit buffer, "0" is written to the DRQI bit in the EO0I status register (EP0IS) to clear it (because the initial value of the PRQI interrupt resource bit is "1"). For confirmation of the normal end of the status stage, "1" is written to the DRQIIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S). Also, the SETP bit and DRQO interrupt resource bit are cleared, and then there is a return from the interrupt.

- Data stage post-processing

  The DRQO bit in the EP0O status register (EP0OS) becomes "1" upon completion of the OUT-direction data stage. CPU interrupt processing begins when the DRQO bit becomes "1". First, the SIZE bit in the EP0O status register (EP0OS) is verified, and DMA is activated with a specified data count corresponding to the received data count, or CPU read is used to read data from the receive buffer. Then, the DRQO interrupt resource bit is cleared, and then there is a return from the interrupt.

- Command termination processing

  The DRQI bit in the EP0I status register (EP0IS) becomes "1" upon completion of the IN-direction status stage. CPU interrupt processing begins when the DRQI bit becomes "1", and the normal end of the status stage is confirmed. Then, the DRQI interrupt resource bit is cleared, and then the processing returns.

31.4.3 STALL Response and Release

For Endpoint0 and For Endpoints 1 to 5, this section explains STALL response and release procedures.

■ STALL response and release procedures for Endpoint0

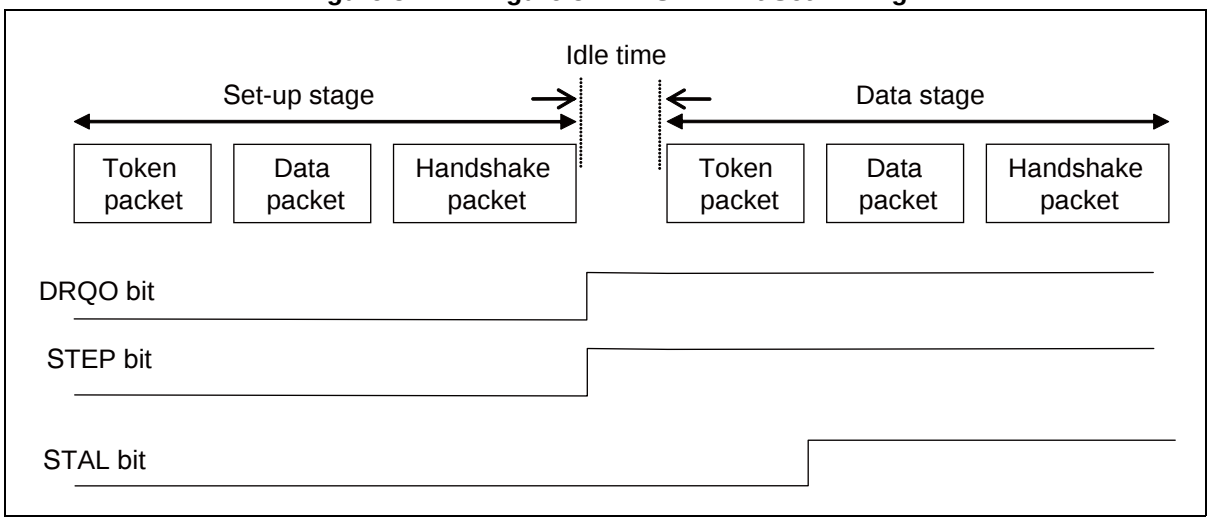
STALL response and release procedures for Endpoint0 are executed with STAL bit of EP0 Control Register (EP0C).

- Set timing of STAL bit

  For STALL response, interprets the command at detecting SETP bit of "1" (DRQO bit = 1 for interrupt) that indicates the set-up stage of control transfer. (See Figure 31.4-7.)

  After setting STAL bit, clear interrupt cause (DRQO bit).

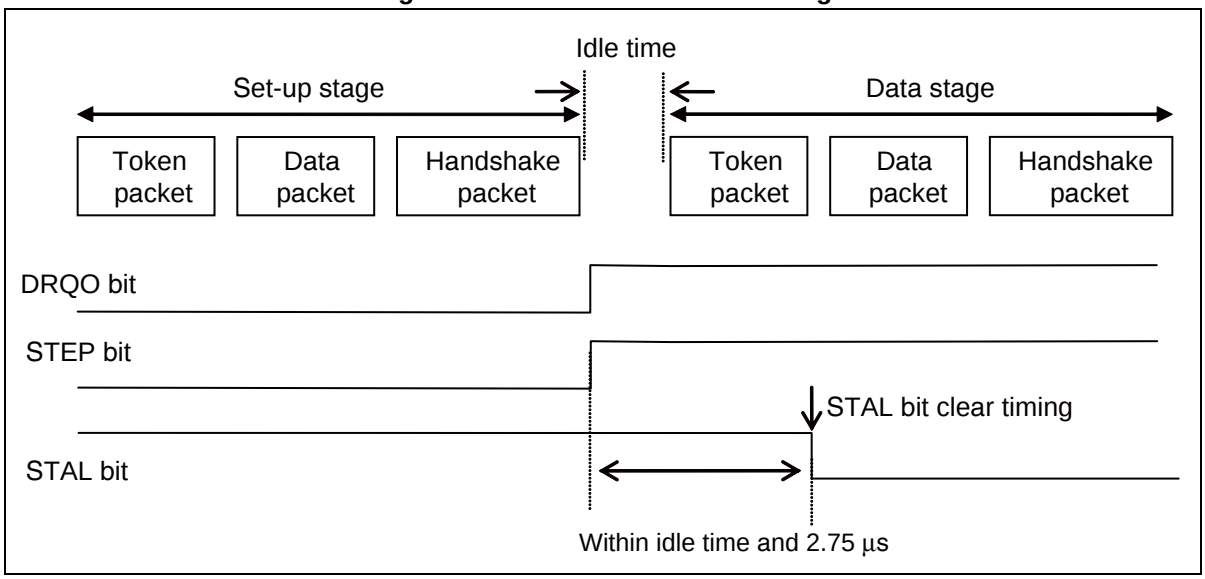
Figure 31.4-7 Figure 31.4-7 STAL Bit Set Timing



- STAL Bit Clear Timing

  For STALL release, clears STAL bit at detecting SETP bit of "1" (DRQO bit = 1 for interrupt) that indicates the set-up stage of control transfer, and sets STAL bit if the STALL response is required. (See Figure 31.4-8.)

Figure 31.4-8 STAL Bit Clear Timing



For STALL response release (STAL bit clear), clear STAL bit the period between the time when STEP bit of "1" (DRQ0 bit= 1 for interrupt) is detected and the time when the data packet transmission/reception of the next data stage is started. The period between the time when DRQ0 becomes "1" and the time when STAL bit is cleared is as follows: (Transfer speed: at Full speed of 12Mbps) When STAL bit is not cleared in the following period, execute STAL response with the handshake of data stage.

The period between the time when DRQ0 BIT of "1" is detected and the time when STAL bit is cleared: within idle time + 2.75 µs

* When idle time is the shortest period of 2-bit transfer time, the above period is within about 2.9 µs.

If the STAL bit clear cannot executed within the above period, take an appropriate countermeasures such as lengthening of the idle time with a driver of USB host.

■ STALL response /release of Endpoints 1 to 5

STALL response /release of Endpoints 1 to 5 are controlled with Control registers of EP1 to EP5 and internal condition bit

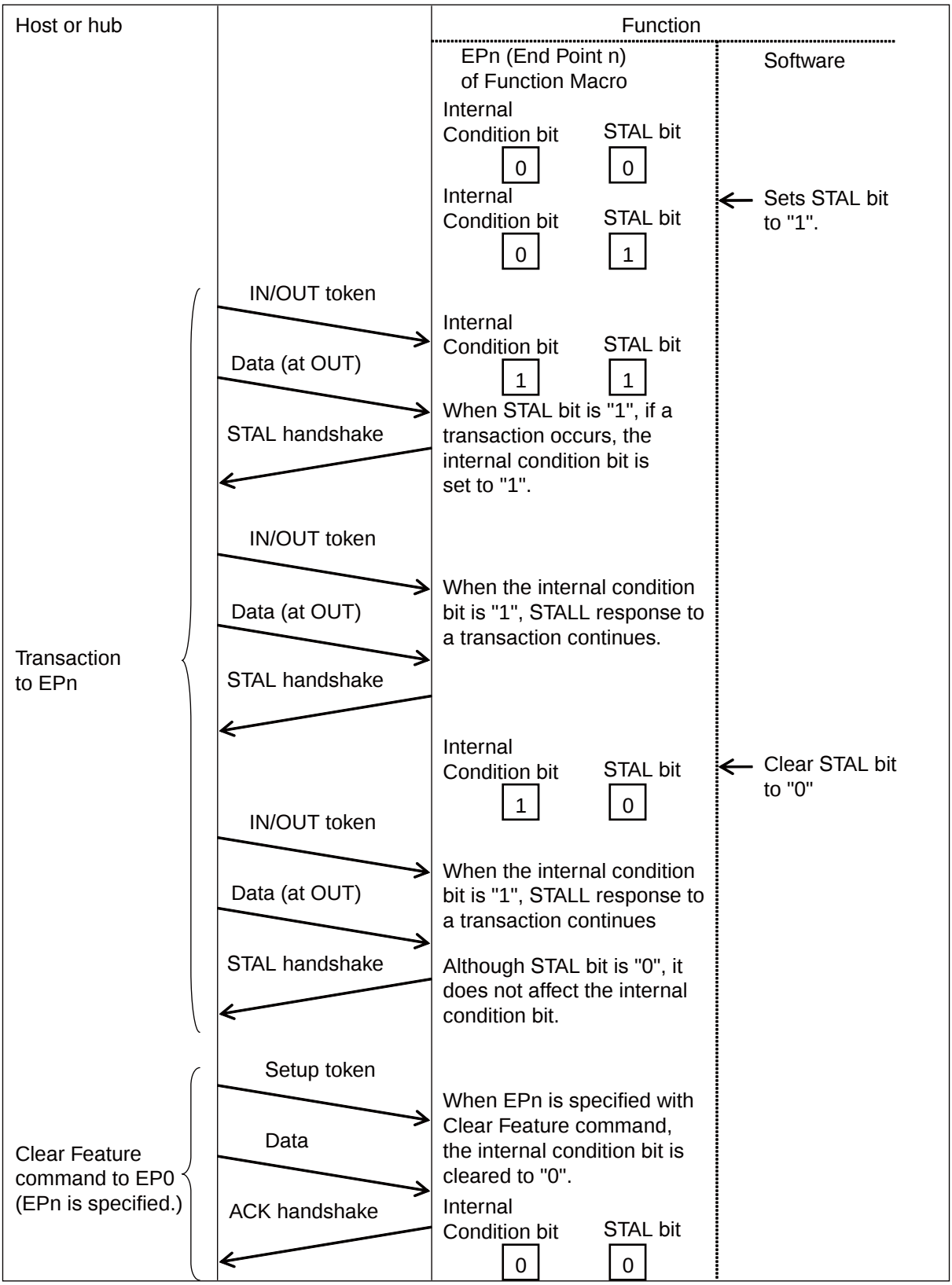
- To execute STAL response with software

  The procedures to execute STAL response with software are shown in Figure 31.4-9. To execute STAL response, set STAL bit of the relevant endpoint with software. In this time, the internal condition bit does not change. Furthermore, when a host generates a transaction to the endpoint where STAL bit is set, hardware would automatically set the internal condition bit of the relevant endpoint and gives STALL response to the host.

  Once the internal condition bit is set, the internal condition bit has been set and continues STAL response until Clear Feature command is issued from the host despite of the clearing of STAL bit.

  As long as the STAL bit is set, STAL bit response continues even if the internal condition bit is cleared with Clear Feature command because the internal condition bit is set every time a transaction to the relevant endpoint occurs. Therefore, to release the STAL response, be sure to clear STAL bit and the internal condition bit with the Clear Feature command.

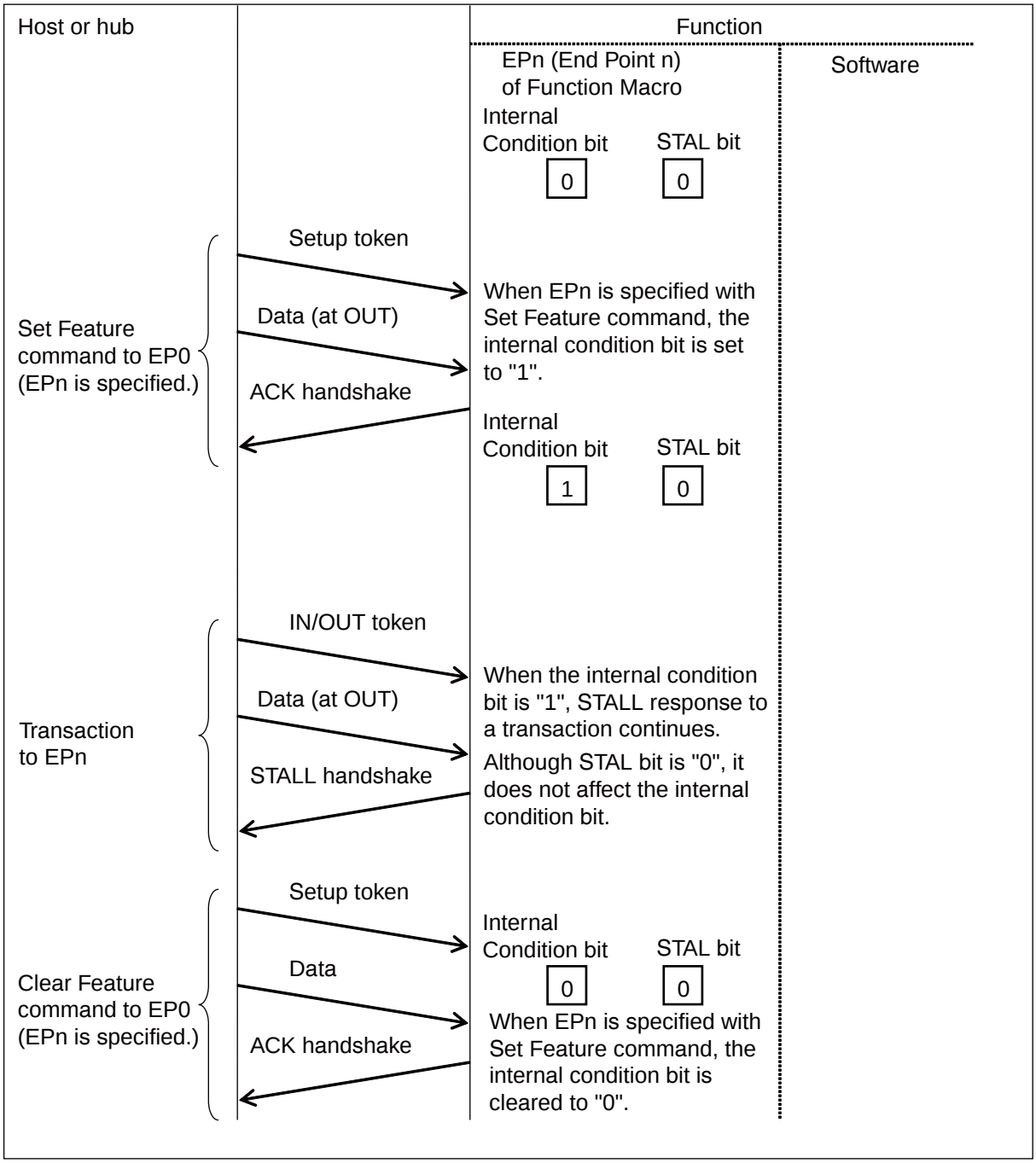
Figure 31.4-9 Case where STALL response is executed with software processing



- To automatically execute STALL response with hardware.

  The procedures to execute STAL response with hardware are shown in Figure 31.4-10. When STALL response is set with Set Feature command, the hardware would automatically set the internal condition bit of the relevant endpoint and gives the STALL response regardless of the STAL bit. Once the internal condition bit is set, the internal condition bit has been held until Clear Feature command is issued from the host to clear the internal condition bit regardless of STAL bit. After the relevant bit is cleared with Clear Feature command, STAL bit is referenced. Therefore, to clear STAL response, be sure to clear the internal condition bit with Clear Feature command.

Figure 31.4-10 Case where STALL response is executed with hardware automatically



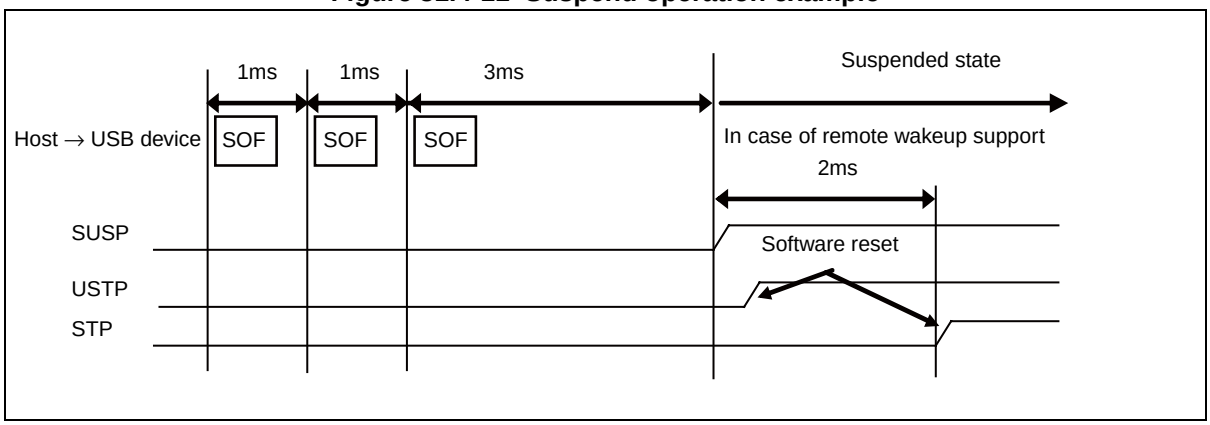
31.4.4 Suspend Function

This section explains the operation that begins at the time of USB device transition to the suspended state and continues until the USB device enters standby mode (watch mode or stop mode). In a bus-powered configuration, set the power dissipation of the USB device in the suspended state at 500 μA or less.

The SUSP bit in the UDC status register (UDCS) becomes "1" when the USB device core detects the suspended state.

Figure 31.4-11 shows the suspend operation.

Figure 31.4-11 Suspend operation example



- Suspend processing

  When no operation has occurred for a period equal to or longer than 3 ms on the USB bus, the USB function detects the suspend condition, the SUSP bit in the UDC status register (UDCS) becomes "1", and the interrupt resource is set. If the device supports remote wakeup, there is a wait for 2 ms from this point in time (preventing remote wakeup during this period), and standby mode (watch mode or stop mode) is set.

31.4.5 Wakeup Function

The USB protocol has the following two methods for the USB device transition from the suspended state to the wakeup state:

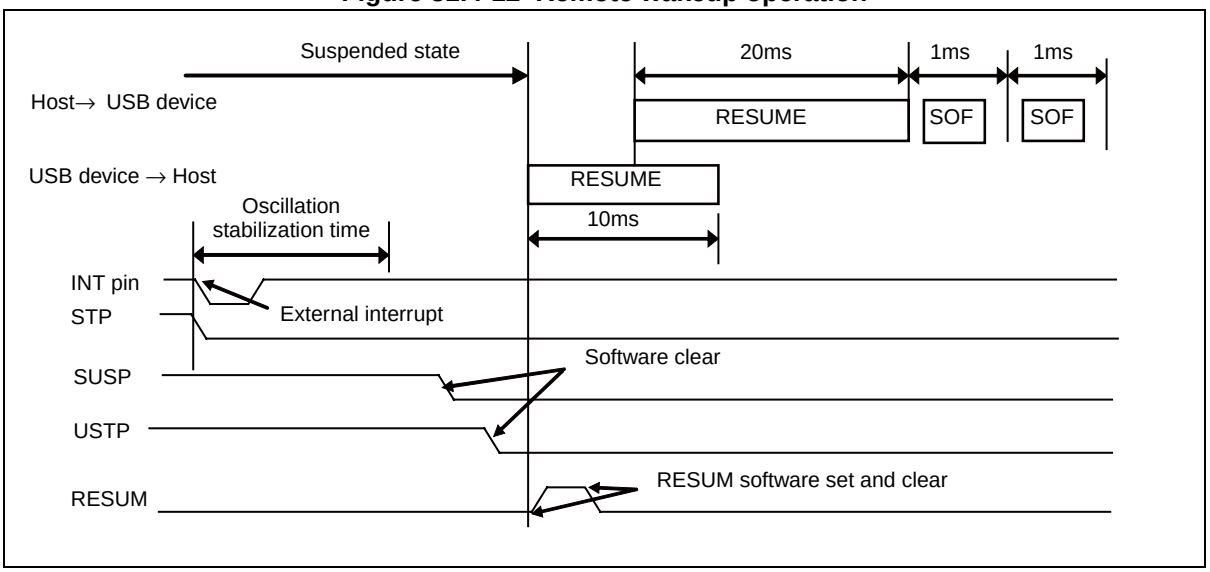
- Remote wakeup by the USB device
- Wakeup by the host

This section explains their operations.

■ Remote wakeup operation

Figure 31.4-12 shows the remote wakeup operation.

Figure 31.4-12 Remote wakeup operation



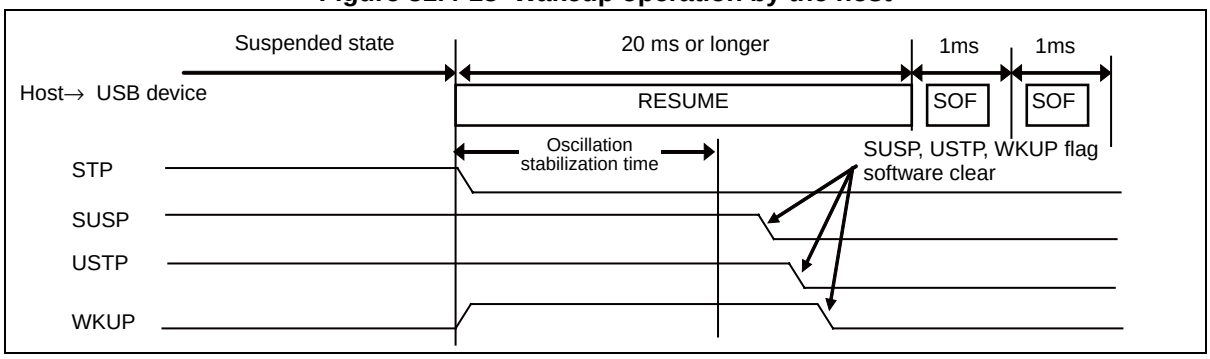
The USB device must perform the following procedure:

1. Use an external interrupt to make the USB device return from stop mode.
2. Write "0" in the SUSP bit of the UDC status register (UDCS) to clear it.
3. Write "0" in the USTP bit of the UDC control register (UDCC) to clear it.
4. Write "1" in the RESUM bit in the UDC control register (UDCC).
5. Write "0" in the RESUM bit in the UDC control register (UDCC) to clear it.

■ Wakeup operation by the host

Figure 31.4-13 shows the wakeup operation by the host.

Figure 31.4-13 Wakeup operation by the host



The USB device must perform the following procedure:

1. Set an oscillation stabilization time that does not exceed 10 ms.
2. Write "0" in the SUSP bit of the UDC status register (UDCS) to clear it.
3. Write "0" in the USTP bit of the UDC control register (UDCC) to clear it.
4. Write "0" in the WKUP bit in the UDC status register (UDCS), which is the interrupt resource, to clear it.

31.4.6 DMA Transfer Function

Data communicated by the USB function can be transferred by DMA between the transmission/ reception buffer and built-in RAM. One of the following two modes can be selected for DMA transfer:

- Packet transfer mode
- Automatic buffer transfer mode

This section explains the two DMA transfer modes.

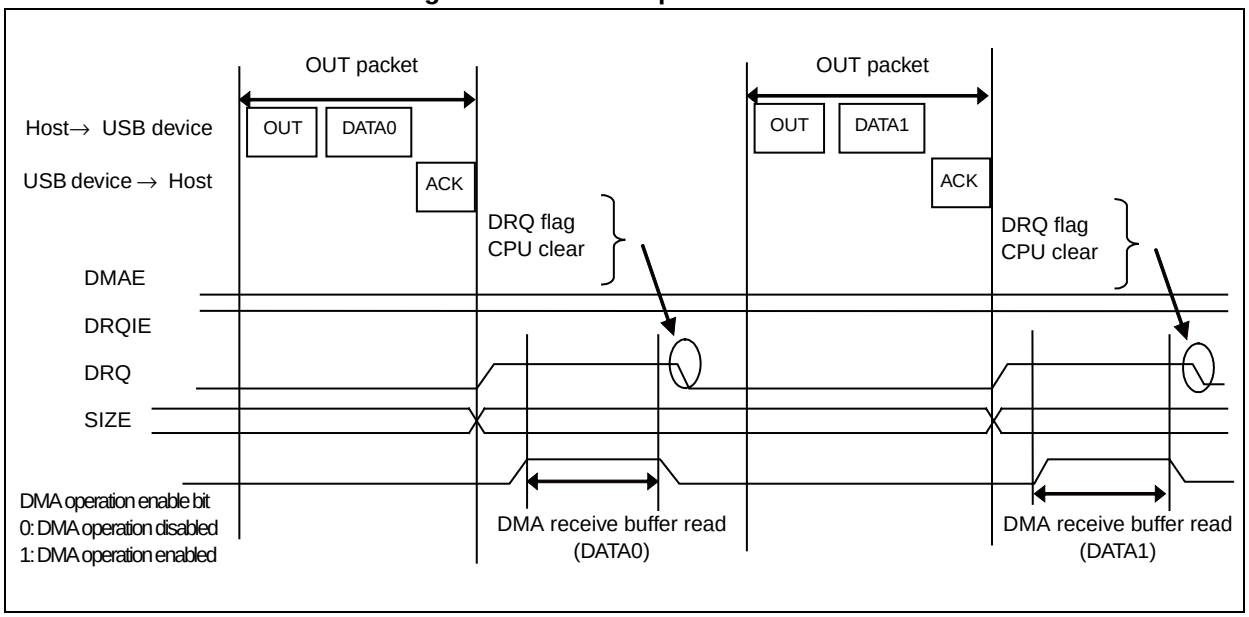
■ Packet transfer mode

In packet transfer mode, the transfer count is set in DMA for each packet, and the interrupt resource is cleared after a DMA transfer is completed. In this mode, the buffers of endpoints 1 to 5 can be accessed. Set DREQ with DREQ selection register (DREQSEL) to use DMA.

● OUT-direction operation

Figure 31.4-14 shows the buffer access timing in an OUT-direction transfer (host to USB device).

Figure 31.4-14 OUT packet transfer



For an OUT-direction transfer, the USB device must perform the following procedures:

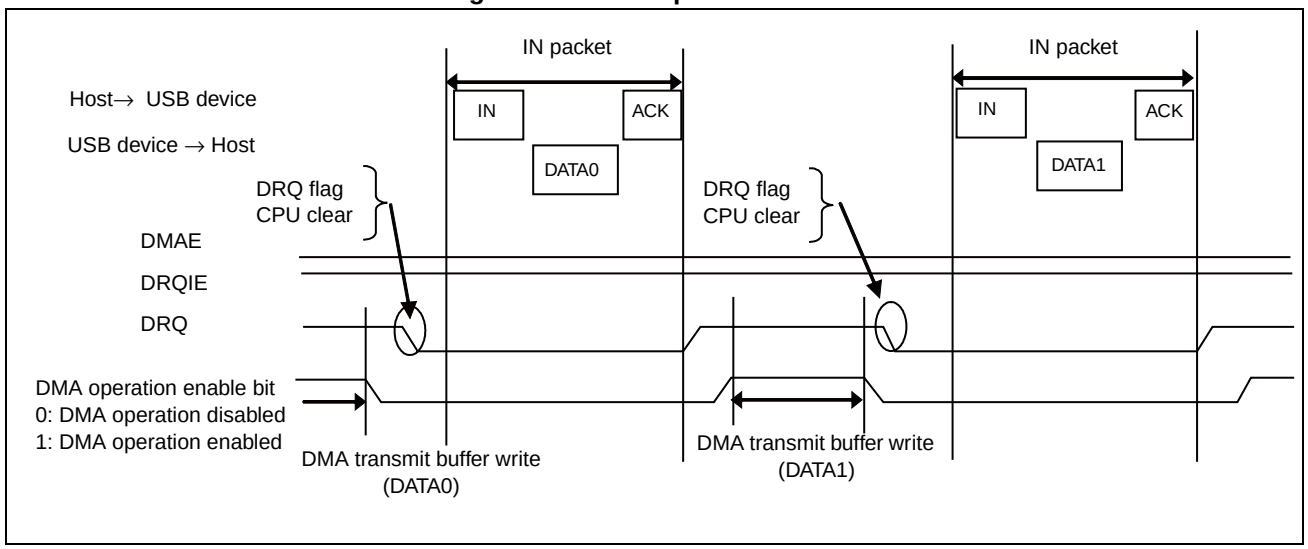
1. Verify the transfer data count after the DRQ flag becomes "1" and interrupt processing begins.
2. Set the block-size-related DMA registers and a transfer count corresponding to the amount of data to be transferred, and enable DMA to start the transfer.
3. Clear the following flags (bits) of registers after the transfer:
   - DRQ bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S)
   - Interrupt resource flag of the relevant register among the DMA channel status registers (DCSR0 to DCSR7) of the DMA controller (DMAC)

   Return from the interrupt processing.

● IN-direction operation

Figure 31.4-15 shows the buffer access timing in an IN-direction transfer (USB device to host).

Figure 31.4-15 IN packet transfer



For an IN-direction transfer, the USB device must perform the following procedure:

1. After the DRQ flag becomes "1" and interrupt processing begins, set the block-size-related DMA registers and a transfer count corresponding to the amount of data to be transferred in the next IN packet, and enable DMA to start the transfer.
2. Clear the following flags (bits) of registers after the transfer:
   - DRQ bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S)
   - Interrupt resource flag of the relevant register among the DMA channel status registers (DCSR0 to DCSR7) of the DMA controller (DMAC)

   Return from the interrupt processing.

■ Automatic buffer transfer mode

In automatic buffer transfer mode, specified data count is transferred at one time according to the setting.

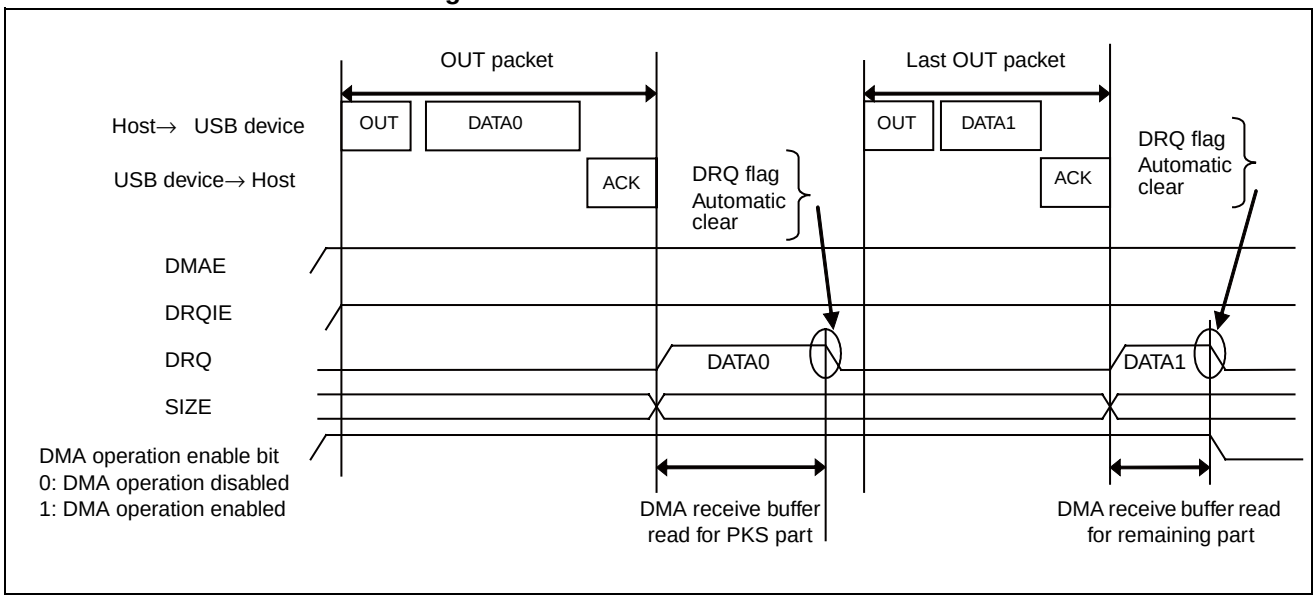
In this transfer mode, even-number bytes can be transferred. To transfer the odd-number bytes, CPU transfer processing is necessary. Before using DMA transfer, set DREQ with DREQ selection register.

Set the total transfer data count in DMA beforehand, and also set the transfer enable bit. If automatic buffer transfer mode is enabled (DMAE = 1) when the DRQ flag is set after a transfer from the host, data count specified by the PKS bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) is transferred. Then, the interrupt resource is automatically cleared. After the transfer from the host, similar processing is repeated as many times as the specified transfer data count set in DMA. During this processing, no setting by the CPU is required, and this mode transfers data at one time according to the setting. To perform the subsequent transfer in a case where a CPU interrupt is entered after the last data transfer, make DMA controller (DMAC) settings again, enable DMA, and return from the interrupt processing. Since the USB function is used with the setting of DMAE = 1 in this mode, only buffer access for endpoints 1 to 5 is valid.

● OUT-direction operation

Figure 31.4-16 shows the buffer access timing in an OUT-direction transfer (host to USB device).

Figure 31.4-16 OUT-direction transfer



For an OUT-direction transfer, the USB device must perform the following procedures:

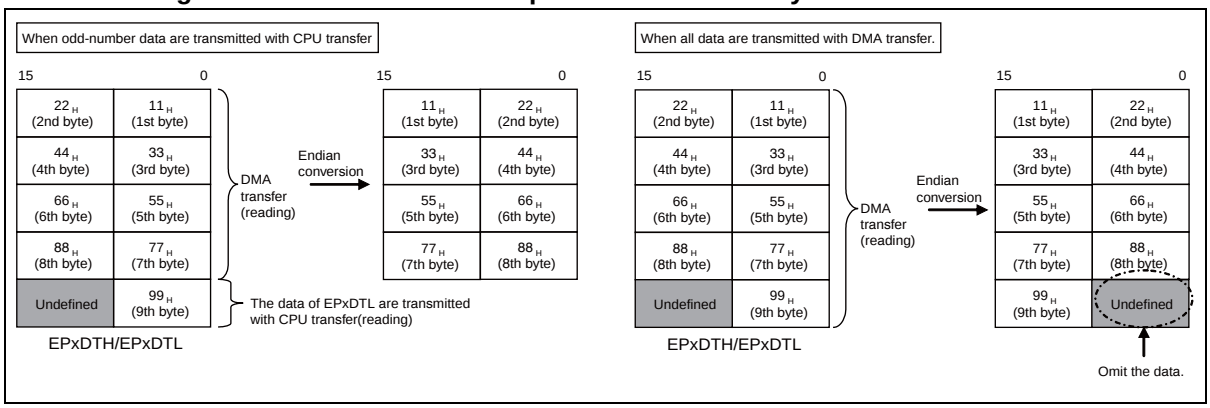
1. Set the block-size-related DMA registers and a transfer count corresponding to the total amount of data to be transferred, and enable DMA to start the transfer.
2. Write "1" in the DMAE bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C) to set automatic buffer transfer mode, and then write "1" in the DRQIE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S) to enable interrupts.
3. Clear the relevant interrupt resource flag in the DMA channel status register (DCSR0 to DCSR7) of the DMA controller (DMAC) after the transfer, and make the DMAC setting again as necessary.

   Return from the interrupt processing.

For transmitting odd-number data with DMA transfer, the following two methods are available.

- For the last data, execute the CPU transfer and read the lower byte (EPxDTL).
- Transmit all data and one byte data together with DMA transfer and omit the last data after endian conversion.

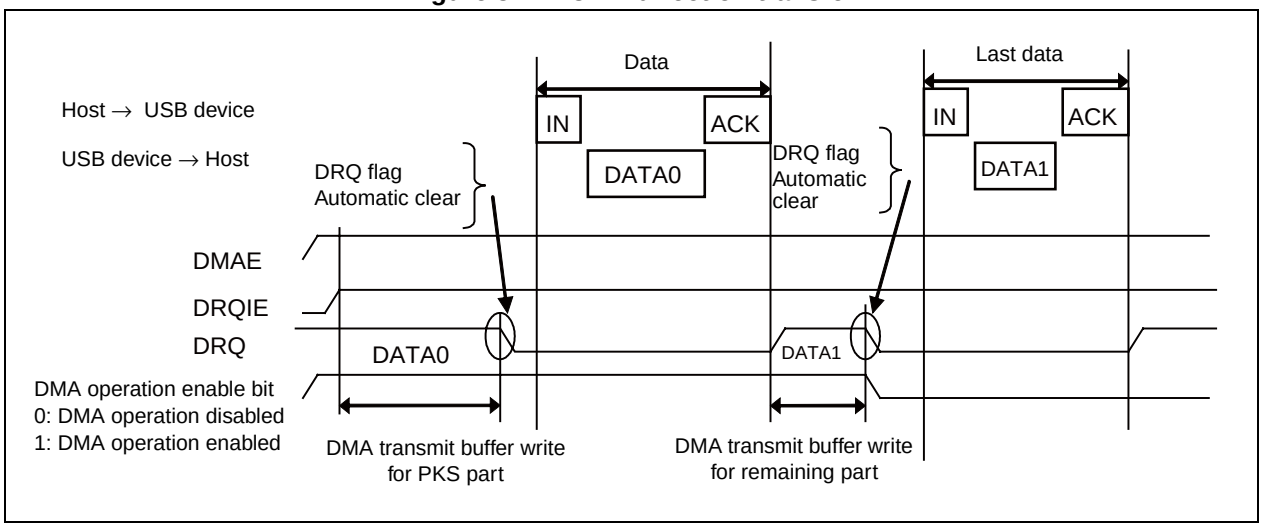
Figure 31.4-17 Transfer Example of Odd-number bytes in OUT direction



● IN-direction operation

Figure 31.4-18 shows the buffer access timing in an IN-direction transfer (USB device to host).

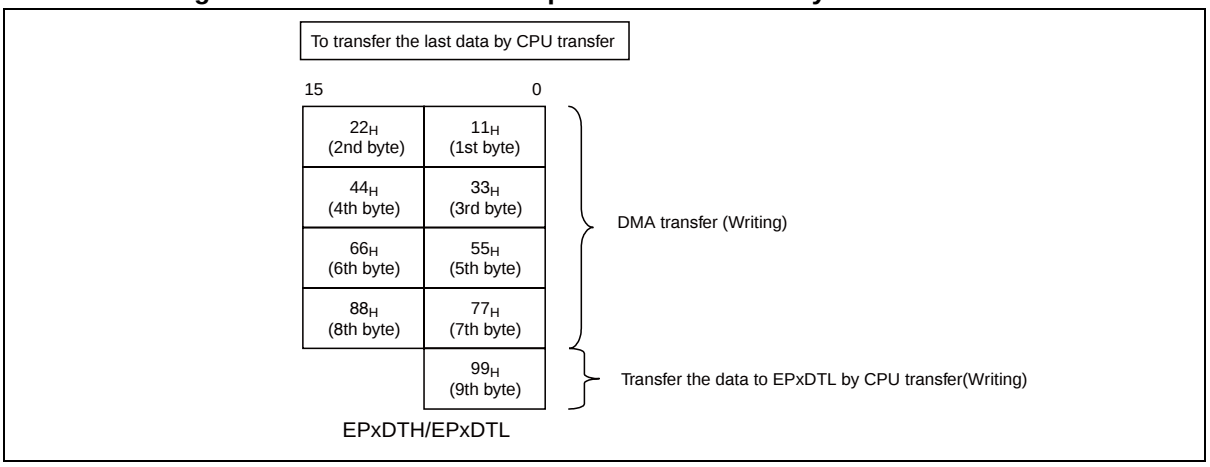
Figure 31.4-18 IN-direction transfer



For an IN-direction transfer, the USB device must perform the same operations as for an OUT-direction transfer. Referring to "OUT-direction operation", execute the processing in the same sequence.

For transmitting odd-number data with DMA transfer, transfer the last data with CPU transfer and write them in the lower byte (EPxDTL).

Figure 31.4-19 Transfer Example of Odd-number Byte in IN Direction

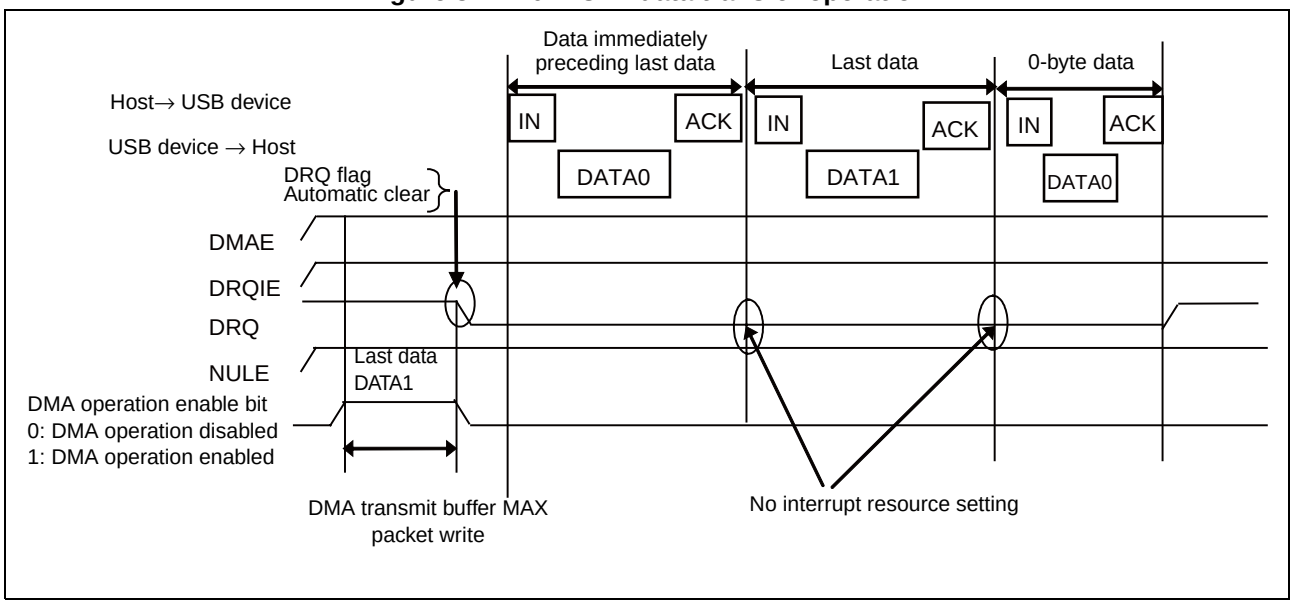


31.4.7 NULL Transfer Function

This section explains NULL transfer mode. In NULL transfer mode, if the number of data bytes transferred from the USB function reaches the max. packet count in the last packet transferred, a 0-byte transferred is automatically performed in the transfer of the next packet.

In NULL transfer mode, a transfer is performed under the following conditions: automatic buffer transfer mode is set (DMAE = 1 in the relevant register among the EP1 to 5 control registers (EP1C to EP5C)), and an IN-direction data transfer request has been generated. Under these conditions, if the DMA data count for the last data write becomes 0 when DMA write with the maximum transfer bytes is completed, a 0-byte data transfer is automatically set upon receipt of the last IN-direction data transfer request from the host. Then, 0-byte data is automatically transmitted for the next IN-direction data transfer request. After the last data is written to the buffer with DMA and until the 0-byte data is read from the host, the DRQ interrupt request flag is not set. Figure 31.4-20 shows the buffer access timing.

Figure 31.4-20 NULL data transfer operation



<Note>

The NULL transfer function is valid only for an IN transfer from the USB device to the host.

The USB device must perform the following procedure:

1. Write "1" in the DMAE bit in the relevant register among the EP1 to 5 control registers (EP1C to EP5C).

   Automatic buffer transfer mode is set.

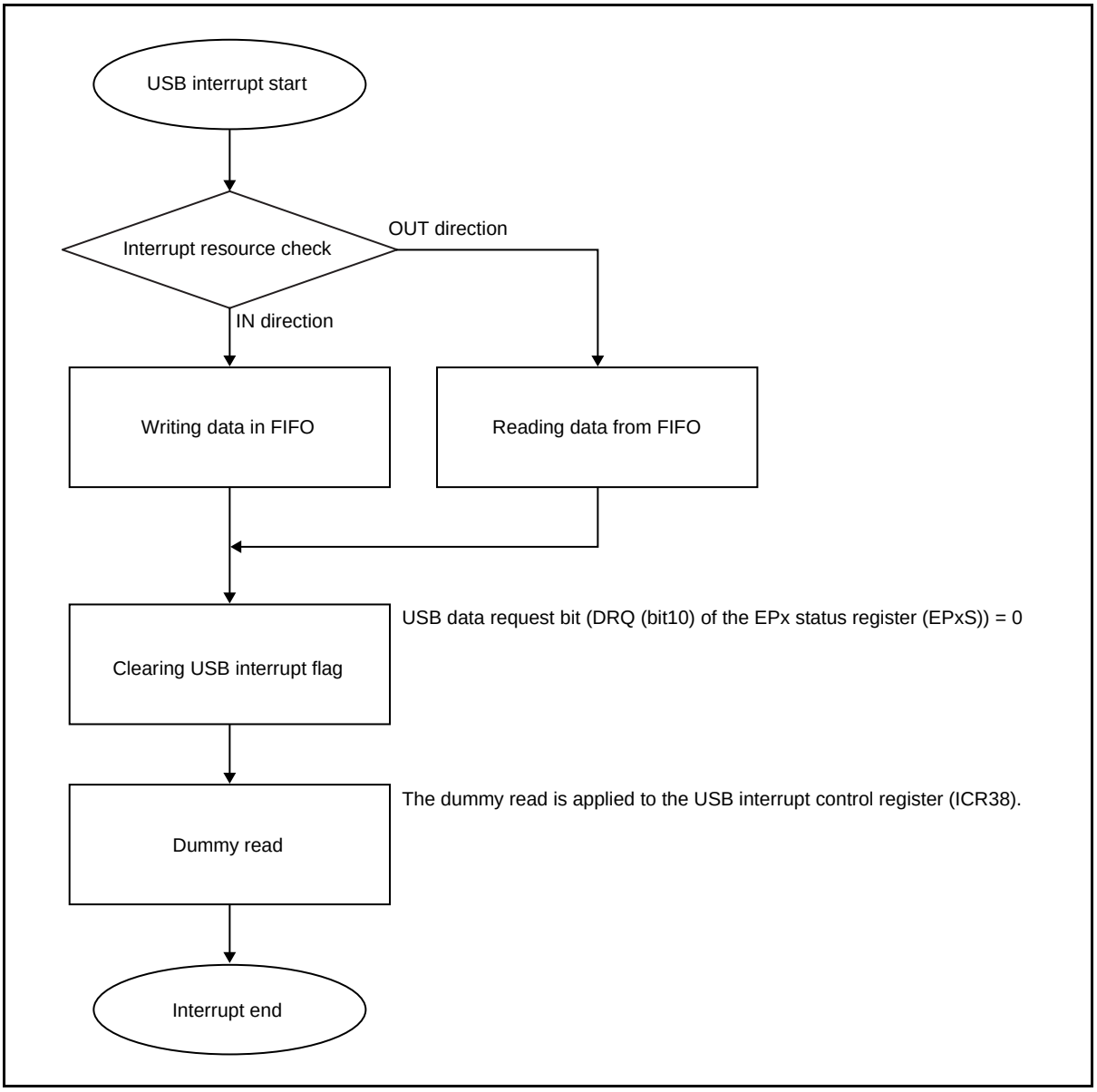
2. Write "1" in the DRQIE bit and NULE bit in the relevant register among the EP1 to 5 status registers (EP1S to EP5S).

   Interrupts are enabled, and NULL automatic transfer mode is set.

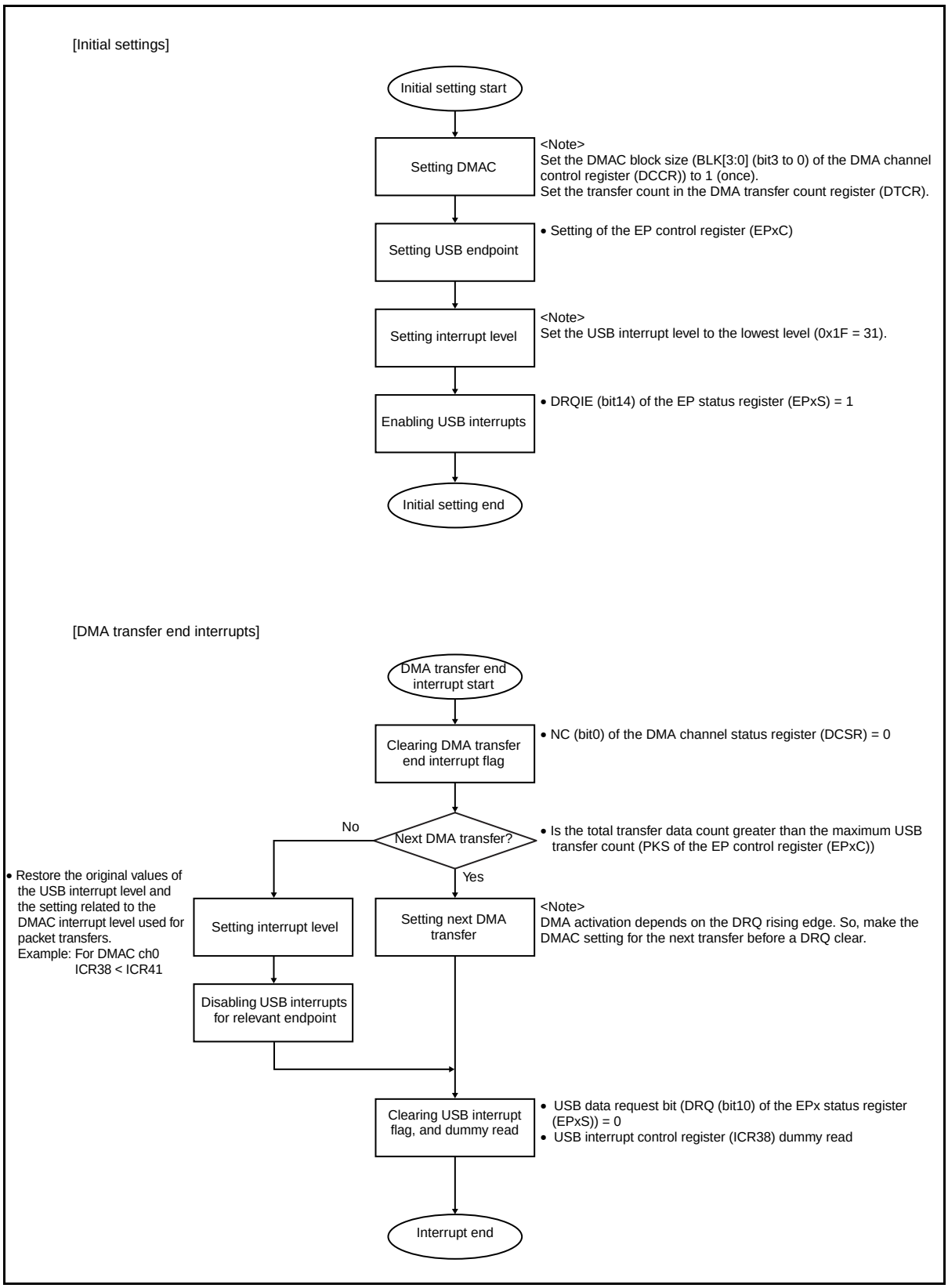
31.4.8 Software Control Examples

This section shows flowcharts for CPU transfers, packet transfer mode (IN/OUT), and automatic buffer transfer mode (IN/OUT).

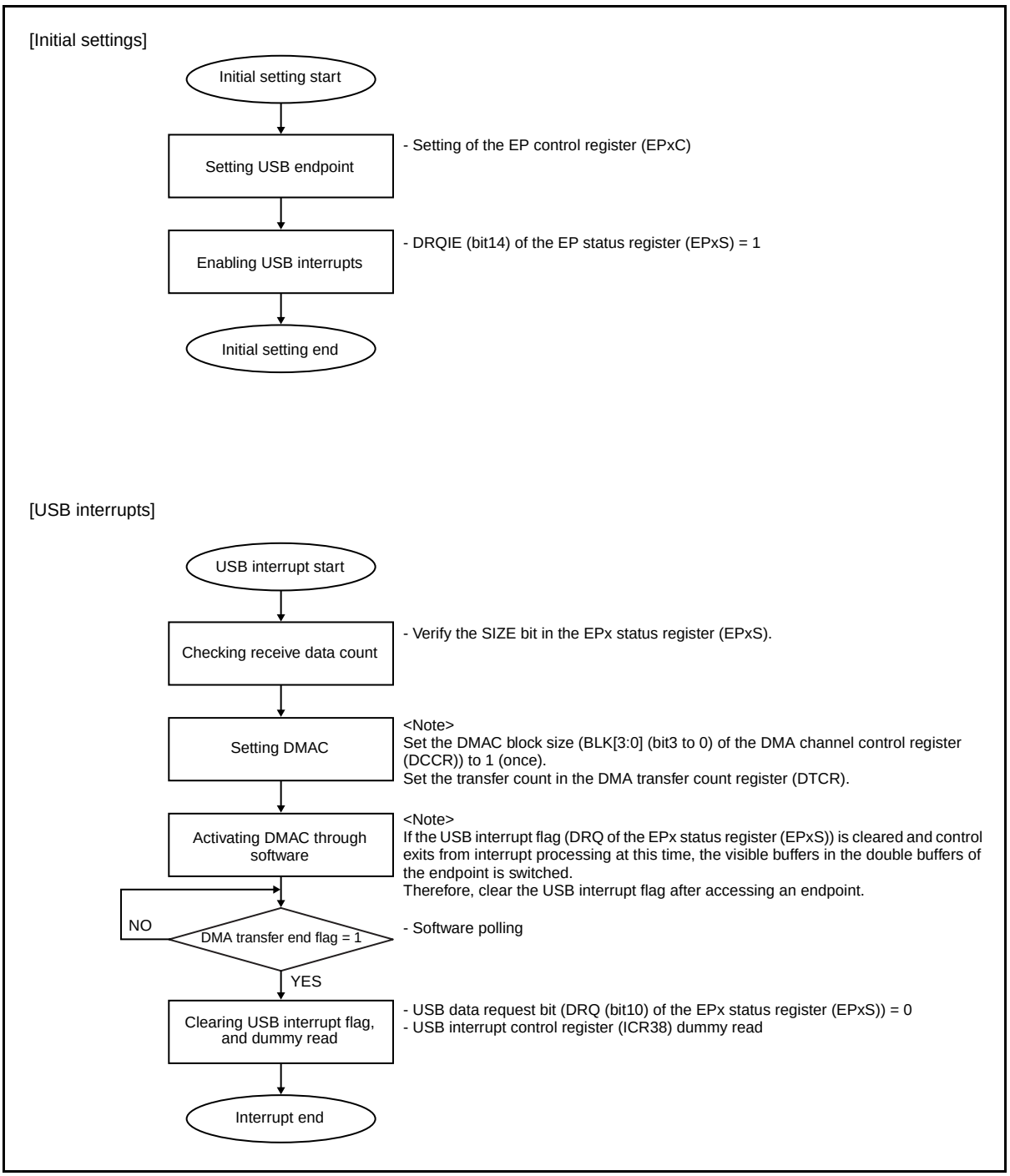
■ Control example for a CPU transfer


USB interrupt start
Interrupt resource check
OUT direction
IN direction
Writing data in FIFO
Reading data from FIFO
Clearing USB interrupt flag
USB data request bit (DRQ (bit10) of the EPx status register (EPxS)) = 0
Dummy read
The dummy read is applied to the USB interrupt control register (ICR38).
Interrupt end

■ Control example for IN-direction packet transfer mode

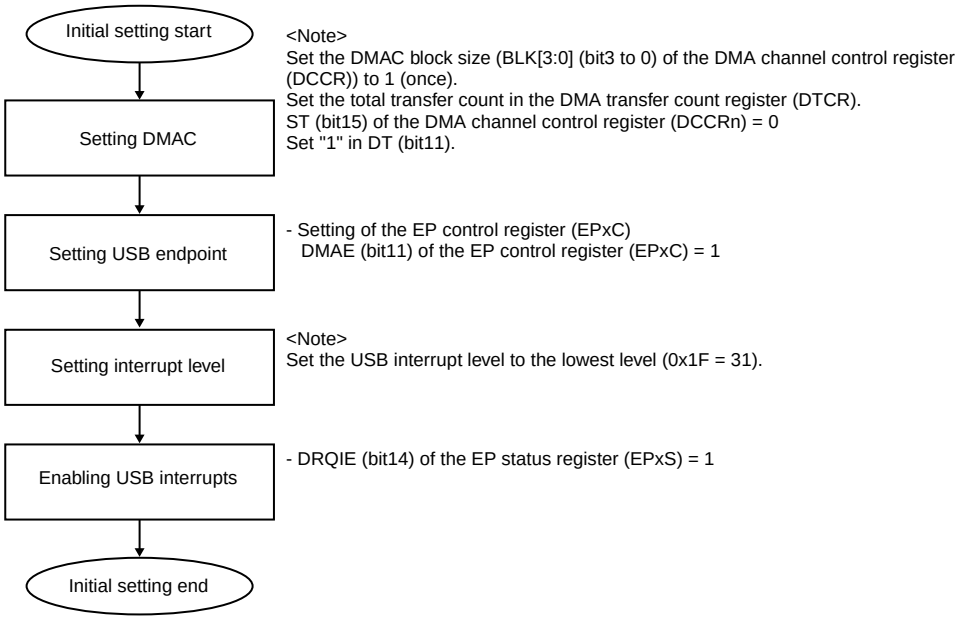

[Initial settings]
Initial setting start
Setting DMAC
<Note>
Set the DMAC block size (BLK[3:0] (bit3 to 0) of the DMA channel control register (DCCR)) to 1 (once).
Set the transfer count in the DMA transfer count register (DTCR).
Setting USB endpoint
• Setting of the EP control register (EPxC)
Setting interrupt level
<Note>
Set the USB interrupt level to the lowest level (0x1F = 31).
Enabling USB interrupts
• DRQIE (bit14) of the EP status register (EPxS) = 1
Initial setting end
[DMA transfer end interrupts]
DMA transfer end interrupt start
Clearing DMA transfer end interrupt flag
• NC (bit0) of the DMA channel status register (DCSR) = 0
Next DMA transfer?
No
Yes
• Is the total transfer data count greater than the maximum USB transfer count (PKS of the EP control register (EPxC))
• Restore the original values of the USB interrupt level and the setting related to the DMAC interrupt level used for packet transfers.
Example: For DMAC ch0
ICR38 < ICR41
Setting interrupt level
Disabling USB interrupts for relevant endpoint
Setting next DMA transfer
<Note>
DMA activation depends on the DRQ rising edge. So, make the DMAC setting for the next transfer before a DRQ clear.
Clearing USB interrupt flag, and dummy read
• USB data request bit (DRQ (bit10) of the EPx status register (EPxS)) = 0
• USB interrupt control register (ICR38) dummy read
Interrupt end

■ Control example for OUT-direction packet transfer mode

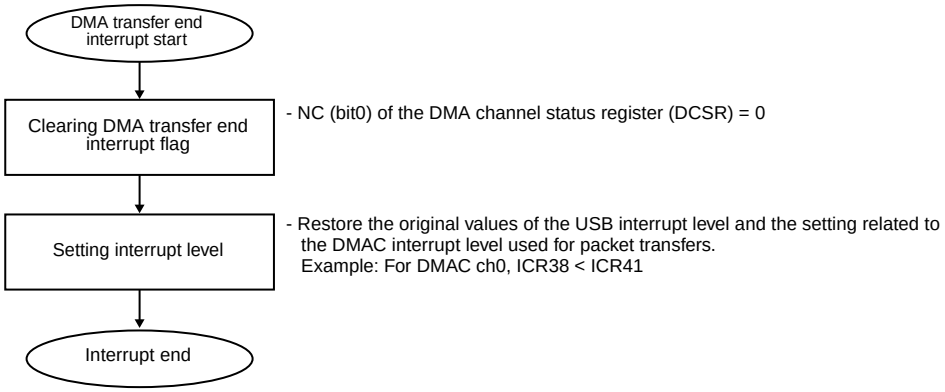

[Initial settings]
Initial setting start
Setting USB endpoint
- Setting of the EP control register (EPxC)
Enabling USB interrupts
- DRQIE (bit14) of the EP status register (EPxS) = 1
Initial setting end
[USB interrupts]
USB interrupt start
Checking receive data count
- Verify the SIZE bit in the EPx status register (EPxS).
Setting DMAC
<Note>
Set the DMAC block size (BLK[3:0] (bit3 to 0) of the DMA channel control register (DCCR)) to 1 (once).
Set the transfer count in the DMA transfer count register (DTCR).
Activating DMAC through software
<Note>
If the USB interrupt flag (DRQ of the EPx status register (EPxS)) is cleared and control exits from interrupt processing at this time, the visible buffers in the double buffers of the endpoint is switched.
Therefore, clear the USB interrupt flag after accessing an endpoint.
NO
DMA transfer end flag = 1
- Software polling
YES
Clearing USB interrupt flag, and dummy read
- USB data request bit (DRQ (bit10) of the EPx status register (EPxS)) = 0
- USB interrupt control register (ICR38) dummy read
Interrupt end

■ Control example for an IN-direction automatic transfer with a specified data count

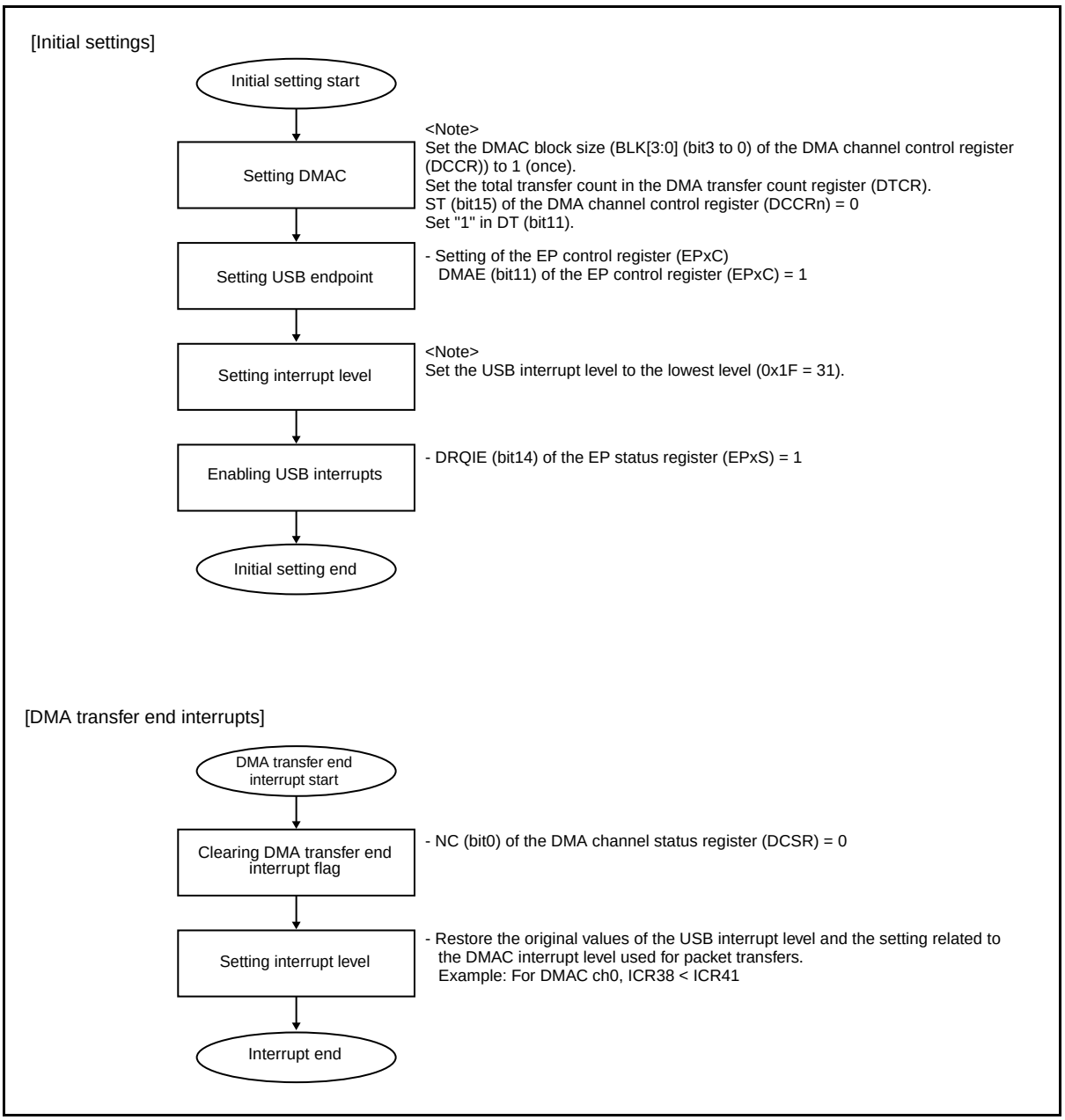
[Initial settings]


Initial setting start
Setting DMAC
Setting USB endpoint
Setting interrupt level
Enabling USB interrupts
Initial setting end
<Note>
Set the DMAC block size (BLK[3:0] (bit3 to 0) of the DMA channel control register (DCCR)) to 1 (once).
Set the total transfer count in the DMA transfer count register (DTCR).
ST (bit15) of the DMA channel control register (DCCRn) = 0
Set "1" in DT (bit11).
- Setting of the EP control register (EPxC)
DMAE (bit11) of the EP control register (EPxC) = 1
<Note>
Set the USB interrupt level to the lowest level (0x1F = 31).
- DRQIE (bit14) of the EP status register (EPxS) = 1

[DMA transfer end interrupts]


DMA transfer end interrupt start
Clearing DMA transfer end interrupt flag
Setting interrupt level
Interrupt end
- NC (bit0) of the DMA channel status register (DCSR) = 0
- Restore the original values of the USB interrupt level and the setting related to the DMAC interrupt level used for packet transfers.
Example: For DMAC ch0, ICR38 < ICR41

■ Control example for an OUT-direction automatic transfer with a specified data count


[Initial settings]
Initial setting start
Setting DMAC
<Note>
Set the DMAC block size (BLK[3:0] (bit3 to 0) of the DMA channel control register (DCCR)) to 1 (once).
Set the total transfer count in the DMA transfer count register (DTCR).
ST (bit15) of the DMA channel control register (DCCRn) = 0
Set "1" in DT (bit11).
Setting USB endpoint
- Setting of the EP control register (EPxC)
DMAE (bit11) of the EP control register (EPxC) = 1
Setting interrupt level
<Note>
Set the USB interrupt level to the lowest level (0x1F = 31).
Enabling USB interrupts
- DRQIE (bit14) of the EP status register (EPxS) = 1
Initial setting end
[DMA transfer end interrupts]
DMA transfer end interrupt start
Clearing DMA transfer end interrupt flag
- NC (bit0) of the DMA channel status register (DCSR) = 0
Setting interrupt level
- Restore the original values of the USB interrupt level and the setting related to the DMAC interrupt level used for packet transfers.
Example: For DMAC ch0, ICR38 < ICR41
Interrupt end

CHAPTER 32 USB Mini-HOST

This chapter explains the functions and operations of USB Mini-HOST.

32.1 Overview

32.2 Configuration

32.3 Registers

32.4 Explanation of Operations and Setting Procedure Examples

32.5 Flowcharts for tokens in operation with USB Mini-HOST

32.1 Overview

USB Mini-HOST is a function that ensures that a minimum of host operations are performed to enable the system to transfer data with USB devices.

■ Overview

USB Mini-HOST has the following features:

- Automatic detection/support of full-speed transmission
- Automatic detection of connection/disconnection of USB devices
- Support for reset transmission to the USB bus
- Support of IN/OUT/SETUP/SOF tokens
- Automatic detection of a handshake packet for IN token processing (except in the STALL state)
- Automatic detection of a handshake packet for OUT token processing
- Support of a maximum packet length of up to 256 bytes
- Support for different types of error (CRC error, toggle error, timeout error, etc.)
- Support for a wakeup function

<Note>

In case of using USB Mini-HOST, use On-chip bus clock (HCLK) with 13 MHz or over.

● Differences from USB HOST

Table 32.1-1 lists the differences between USB HOST and USB Mini-HOST.

Table 32.1-1 Differences between USB HOST and USB Mini-HOST (1 / 2)

		HOST	Mini-HOST
HUB support		O	X
Transfer	Bulk transfer	O	O
	Control transfer	O	O
	Interrupt transfer	O	O
	Isochronous transfer	O	X
Transfer speed	Low speed	O	X
	Full speed	O	O
Support of PRE packets		O	X
Support of SOF packets		O	O

Table 32.1-1 Differences between USB HOST and USB Mini-HOST (2 / 2)

		HOST	Mini-HOST
Errors	CRC error	O	O
	Toggle error	O	O
	Timeout	O	O
	Maximum size of packet < Received data	O	O
Detection of connection/disconnection of USB devices		O	O
Detection of transfer speed		O	O

O: Supported
X: Not supported

32.2 Configuration

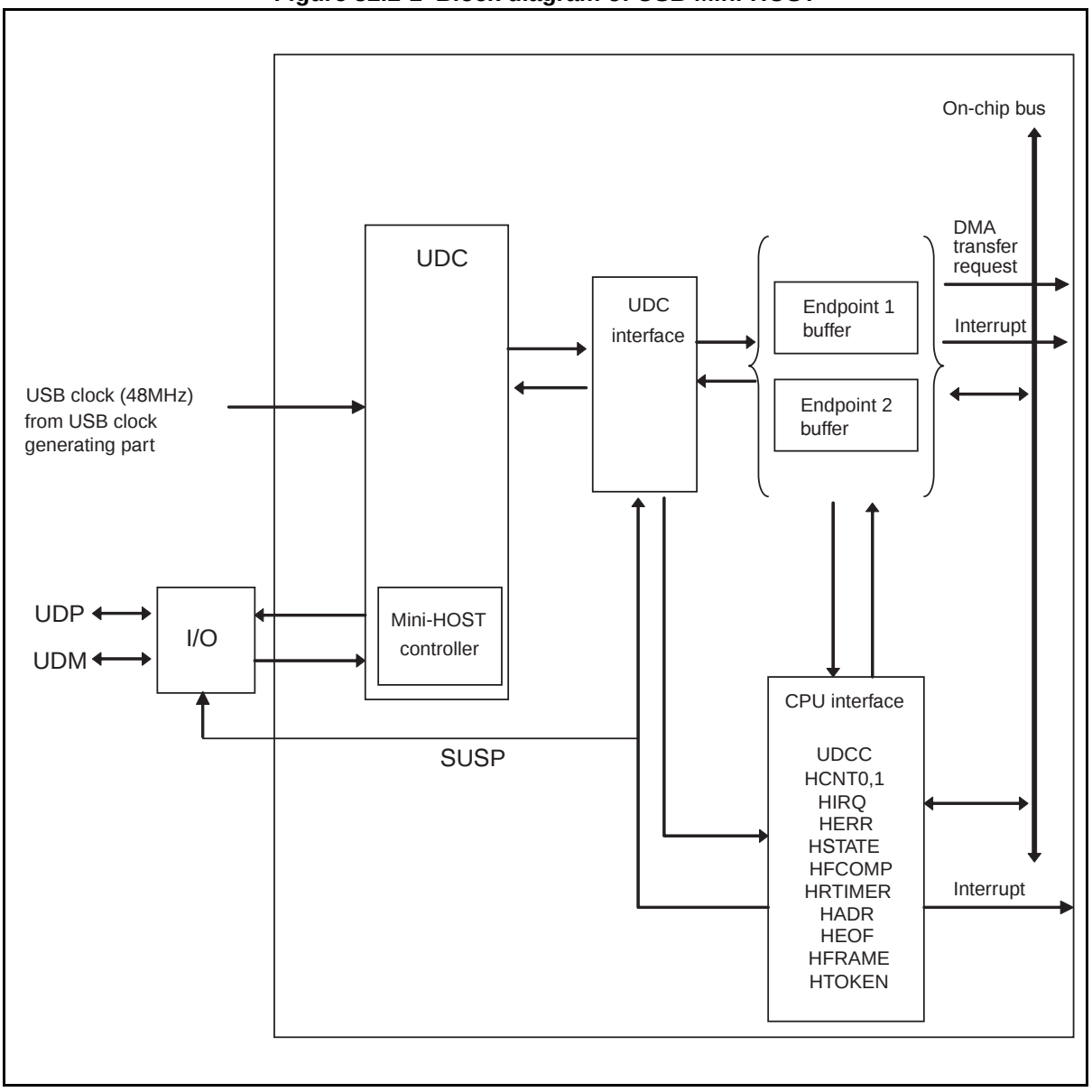
USB Mini-HOST consists of the following blocks:

- CPU I/F
- Buffer
- UDC I/F
- USB bus reset control part
- Reception control part
- Transmission control part
- Host transmission control part

■ Block diagram of USB Mini-HOST

Figure 32.2-1 is a block diagram of USB Mini-HOST.

Figure 32.2-1 Block diagram of USB Mini-HOST



32.3 Registers

This section explains the configuration and functions of registers used by the USB Mini-HOST.

■ USB Mini-HOST register list

Table 32.3-1 lists the registers of USB Mini-HOST.

Table 32.3-1 USB Mini-HOST registers

Abbreviated Register Name	Register Name	Reference
UCCR	USB clock setting register	29.3.1
USBSEL	USB selection register	31.3.1
USBEN	USB enable register	31.3.2
UDCC	UDC control register	31.3.3
EP1C	EP1 control register	31.3.5
EP2C	EP2 control register	31.3.5
EP1S	EP1 status register	31.3.11
EP2S	EP2 status register	31.3.11
EP1DTH	EP1 data register upper	31.3.12
EP1DTL	EP1 data register lower	31.3.12
EP2DTH	EP2 data register upper	31.3.12
EP2DTL	EP2 data register lower	31.3.12
HCNT0	Host control register 0	32.3.1
HCNT1	Host control register 1	32.3.1
HIRQ	Host interrupt register	32.3.2
HERR	Host error status register	32.3.3
HSTATE	Host status register	32.3.4
HFCOMP	SOF interrupt FRAME comparison register	32.3.5
HRTIMER	Retry timer setting register	32.3.6
HADR	Host address register	32.3.7
HEOF	EOF setting register	32.3.8
HFRAME	FRAME setting register	32.3.9
HTOKEN	Host token endpoint register	32.3.10

■ List of register bits update timing UDCC. RST dependence

Table 32.3-2 List of register bits update timing UDCC. RST dependence

	Register	Bit
Register bits that have to be updated when UDCC.RST=1	HCNT0	HOST
	HSTATE	CLKSEL
	EP1C	EPEN, TYPE, DIR, PKS1
	EP2C	EPEN, TYPE, DIR, PKS2
Register bits that are initialized when UDCC.RST=1. (Update these bits when UDCC.RST=0.)	HCNT0	URST
	HIRQ	TCAN, RWKIRQ, URIRQ, CMPIRQ, CNNIRQ, DIRQ, SOFIRQ
	HERR (all bits)	LSTSOF, RERR, TOUT, CRC, TGERR, STUFF, HS
	HSTATE	SOFBUSY, SUSP
	HFRAME	FRAME0, FRAME1
	HTOKEN (all bits)	TGGL, TKNEN, ENDPT
	EP1S	BFINI, DRQ, SPK
	EP2S	BFINI, DRQ, SPK
Register bits that are not affected by UDCC.RST.	HCNT0	RWKIRE, URIRE, CMPIRE, CNNIRE, DIRE, SOFIRE
	HCNT1	SOFSTEP, CANSEL, RETRY
	HIRQ	CNNIRQ, DIRQ
	HFCOMP	HFRAMECOMP
	HSTATE	TMODE, CSTAT
	HRTIMER0 to HRTIMER2	RTIMER0, RTIMER1, RTIMER2
	HADR	Address
	HEOF	EOF0, EOF1

32.3.1 Host Control Register 0, 1 (HCNT)

This register sets the USB operation mode and interrupts.

Figure 32.3-1 shows the bit configuration of host control register 0, 1 (HCNT).

Figure 32.3-1 Bit configuration of host control register 0, 1 (HCNT)

Host Control Register 1

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	Reserved	SOFSTEP	CANCEL	RETRY
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	1
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×	×	×	×	×	×	×	×

Host Control Register 0

bit	7	6	5	4	3	2	1	0
	RWKIRE	URIRE	CMPIRE	CNNIRE	DIRE	SOFIRE	URST	HOST
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×	×	×	×	×	×	❍	×

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15 to bit11]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).

[bit10]: SOFSTEP (SOF STEP bit)

This bit is a SOF interrupt generation select bit.

If the bit is set to "1", an interrupt flag for SOF (HIRQ.SOFIRQ) is set to "1" every time that SOF processing is executed.

If the bit is set to "0", the setting value of the SOF interrupt FRAME comparison register (HFCOMP) is compared with the lower 8-bit of the frame number for SOF, if the comparison result is a match, the interrupt flag for SOF (HIRQ.SOFIRQ) is set to "1".

Written Value	Explanation
0	Generates an interrupt request according to the HFCOMP setting.
1	Generates an interrupt request during SOF processing.

<Notes>

- If SOF token (TKNEN="001_B") is processed according to the host token end point register (HTOKEN), the interrupt flag for SOF (HIRQ. SOFIRQ) does not become "1" not depending on the setting of this register.
- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit9]: CANCEL (token CANCEL enable bit)

This bit is a token cancellation enable bit.

If "1" is set to this bit and writing a token to be processed in the host token end point register (HTOKEN) in the EOF area (area is set by an EOF setting register) cancels the process of the token. If "0" is set to this bit, writing the token to be processed does not cancel the process of the token. Cancellation of the token can be known by reading the TCAN bit in the host interrupt register (HIRQ).

Written Value	Explanation
0	Continues with the token.
1	Cancels the token.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit8]: RETRY (RETRY enable bit)

This bit is a RETRY enable bit.

If an error such as the NAK error or other error (such as HERR. RERR="1", HERR. TOUT="1", HERR. CRC="1", HERR. TGERR="1" or HERR. STUFF="1") occurs and this bit is set to "1", the operation is retried. The operation is retried for the period specified in the retry timer register (HRTIMER).

Written Value	Explanation
0	Does not retry the operation.
1	Retries the operation.

<Note>

The bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit7]: RWKIRE (Remote WaKe up Interrupt Request Enable bit)

This bit is a resume interrupt enable bit.

If "1" is set to this bit and changing the RWKIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit and changing the RWIRQ bit in host interrupt register (HIRQ) to "1" does not generate an interrupt.

Written Value	Explanation
0	Disables interrupts after a reactivation.
1	Enables interrupts after a reactivation.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit6]: URIRE (Usb bus Rest Interrupt Request Enable bit)

This bit is a bus reset interrupt enable bit.

If "1" is set to this bit and changing the URIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit and changing the URIRQ bit in host interrupt register (HIRQ) to "1" does not generate an interrupt.

Written Value	Explanation
0	Disables interrupts after a USB bus reset.
1	Enables interrupts after a USB bus reset.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit5]: CMPIRE (CoMPletion Interrupt Request Enable bit)

This bit is a token completion interrupt enable bit.

If "1" is set to this bit, changing the CMPIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the CMPIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts at the completion time of a token.
1	Enables interrupts at the completion time of a token.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit4]: CNNIRE (CoNNection Interrupt Request Enable bit)

This bit is a device connection detection interrupt enable bit.

If "1" is set to this bit, changing the CNNIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the CNNIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when a USB device is connected.
1	Enables interrupts when a USB device is connected.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit3]: DIRE (Disconnection Interrupt Request Enable bit)

This bit is a device disconnection detection interrupt enable bit.

If "1" is set to this bit, changing the DIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the DIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when a USB device is disconnected.
1	Enables interrupts when a USB device is disconnected.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit2]: SOFIRE (Start Of Frame Interrupt Request Enable bit)

This bit is a SOF interrupt enable bit.

If "1" is set to this bit, changing the SOFIRQ bit in host interrupt register (HIRQ) to "1" generates an interrupt. If "0" is set to this bit, changing the SOFIRQ bit in host interrupt register (HIRQ) to "1" does not generate the interrupt.

Written Value	Explanation
0	Disables interrupts when SOF is transmitted.
1	Enables interrupts when SOF is transmitted.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

[bit1]: URST (Usb bus ReSeT bit)

This bit is a bus reset bit.

If "1" is set to this bit, a bus reset is processed. "1" is indicated by the bit during the reset operation, and it changes to "0" when the reset is completed. Setting "0" to this bit does not operate anything.

Written Value	Explanation
0	Keeps the USB bus in its current state.
1	Resets the USB bus.

<Notes>

- Nothing is operated by setting "1" to this bit while the RST bit in the UDC control register (UDCC) is "1".
- Do not write "1" to this bit when the SUSP bit in the host status register (HSTATE) is "1" or when processing of a token is in progress.
- Do not update the host control registers (HCNT0, HCNT1) while this bit is "1".

[bit0]: HOST (HOST mode bit)

This bit is a host mode bit.

If "1" is set to this bit, the device is operated in Mini-HOST mode. If "0" is set to this bit, the device is operated in function mode.

Written Value	Explanation
0	Function mode is set.
1	Host mode is set.

<Notes>

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Update this bit only while the RST bit in the UDC control register (UDCC) is "1".
- If operation mode is switched by this bit, it cannot be switched soon. To verify whether the mode is switched by reading this bit.
- To switch from host mode to function mode, verify that the below conditions are satisfied and set "1" to the RST bit in the UDC control register (UDCC) before switching.
  - The SOFBUSY bit in the host status register (HSTATE) is "0".
  - The TKNEN bit in the host token endpoint register (HTOKEN) is "000_B".
  - The SUSP bit in the host status register (HSTATE) is "0".
- To switch from function mode to host mode, and set on the pull-up resistor disconnection side of the general port for pull-up resistor control to disconnect the host or hub.

32.3.2 Host Interrupt Register (HIRQ)

This register indicates the interrupt request flags for USB Mini-HOST. Except the TCAN bit, these bits enable generation of interrupt requests according to the settings of the interrupt enable bits in host control registers 0 and 1 (HCNT0, HCNT1).

Figure 32.3-2 shows the bit configuration of the host interrupt select register (HIRQ).

Figure 32.3-2 Bit configuration of the host interrupt register (HIRQ)

Host Interrupt Register (HIRQ)

bit	7	6	5	4	3	2	1	0
	TCAN	Reserved	RWKIRQ	URIRQ	CMPIRQ	CNNIRQ	DIRQ	SOFIRQ
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍	❍	❍	❍	×	×	❍

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7]: TCAN (Token CANcel flag bit)

This bit is a token cancel flag.

This bit indicates that the processing of token was canceled according to the CANCEL bit in the host control register 1 (HCNT1) when this bit is changed to "1". If "0" is written to this bit, it indicates that the processing of the token wan not canceled. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TCAN	In case of reading	In case of writing
0	No such token has been canceled.	Clears this bit.
1	Such a token has been canceled.	Ignored.

<Notes>

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- No interrupt request is generated with this bit. To process with an interrupt, verify whether the token was cancelled during the interrupt process with SOF.

[bit6]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).

[bit5]: RWKIRQ (Remote WaKe up Interrupt ReQuest bit)

This bit is a remote wake up completion flag.

When this bit changes to "1", it indicates that the remote operation has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the RWKIRE bit in host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

RWKIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by a reactivation.	Clears interrupt factor.
1	An interrupt request has been generated by a reactivation.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit4]: URIRQ (Usb bus Reset Interrupt ReQuest bit)

This bit is a bus reset completion flag.

When this bit changes to "1", it indicates that a reset to a USB bus has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the URIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

URIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by a USB bus reset.	Clears interrupt factor.
1	An interrupt request has been generated by a USB bus reset.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit3]: CMPIRQ (CoMPletion Interrupt ReQuest bit)

This bit is a token completion flag.

When this bit changes to "1", it indicates that a token has been completed. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the CMPIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

CMPIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the completion of a token.	Clears interrupt factor.
1	An interrupt request has been generated by the completion of a token.	Ignored.

<Notes>

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The bit does not change to "1" when the TCAN bit in the host interrupt register (HIRQ) changes to "1".

[bit2]: CNNIRQ (CoNNection Interrupt ReQuest bit)

This bit is a device connection detection flag.

When this bit changes to "1",it indicates that connection of the device has been detected. Changing this bit to "0" does not mean anything. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the CNNIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

CNNIRQ	In case of reading	In case of writing
0	No interrupt request is generated by the detection of the connection with a USB device.	Clears interrupt factor.
1	An interrupt request has been generated by the detection of the connection with a USB device.	Ignored.

<Notes>

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The connection of the device is detected in function mode also.

[bit1]: DIRQ (Disconnection Interrupt ReQuest bit)

This bit is a device disconnection detection flag.

When this bit changes to "1", it indicates that disconnection of the device has been detected. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the DIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

DIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the detection of the disconnection of a USB device.	Clears interrupt factor.
1	An interrupt request has been generated by the detection of the disconnection of a USB device.	Ignored.

<Notes>

- This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".
- The disconnection of the device is detected in function mode also.

[bit0]: SOFIRQ (Start Of Frame Interrupt ReQuest bit)

This bit is a SOF starting flag.

When this bit changes to "1", it indicates that the SOF token process has been started. It does not mean anything if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

While the SOFIRE bit in the host control register 0 (HCNT0) is "1", changing this bit to "1" generates an interrupt request.

SOFIRQ	In case of reading	In case of writing
0	No interrupt request has been generated by the start of an SOF token.	Clears interrupt factor.
1	An interrupt request has been generated by the start of an SOF token.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

32.3.3 Host Error Status Register (HERR)

This register indicates whether an error occurred during data transmission/reception in host mode.

Figure 32.3-3 shows the bit configuration of the host error status register (HERR).

Figure 32.3-3 Bit configuration of the host error status register (HERR)

bit	15	14	13	12	11	10	9	8
	LSTSOF	RERR	TOUT	CRC	TGERR	STUFF	HS	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	11	
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍	❍	❍	❍	❍	❍	

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15]: LSTSOF (LoST SOF bit)

This bit is a lost SOF flag.

When this bit changes to "1", and trying to process the SOF token in host mode, it indicates that the SOF token cannot be processed because processing of another token was in progress. It indicates that a lost SOF error has not been detected if this bit is "0". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

LSTSOF	In case of reading	In case of writing
0	No SOF processing error.	Clears this bit.
1	SOF processing error.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit14]: RERR (Receive Error bit)

This bit is a reception error flag.

When this bit changes to "1", it indicates that the received data is greater than the specified maximum number of packets in host mode. If the reception error is detected, TOUT bit of this register also changes to "1". If this bit is "0", it indicates that the error has not occurred. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

RERR	In case of reading	In case of writing
0	No reception error has occurred.	Clears this bit.
1	A maximum packet reception error has occurred.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit13]: TOUT (Time OUT error bit)

This bit is a timeout error flag.

When this bit changes to "1", it indicates that no response to a token was obtained from the device within the given period in host mode. If this bit is "0", it indicates that the timeout error has not occurred. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TOUT	In case of reading	In case of writing
0	No timeout has occurred.	Clears this bit.
1	A timeout has occurred.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit12]: CRC (CRC error bit)

This bit is a CRC error flag.

When this bit changes to "1", it indicates that a CRC error has been detected in host mode. If this bit is "0", it indicates that the CRC error has not been detected. If a CRC error has occurred, TOUT bit of this register also changes to "1". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

CRC	In case of reading	In case of writing
0	No CRC error has occurred.	Clears this bit.
1	A CRC error has occurred.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit11]: TGERR (ToGgle ERRor bit)

This bit is a toggle error flag.

When this bit changes to "1", it indicates that the received toggle was not a match in host mode. If this bit is "0", it indicates that the toggle error has not been detected. When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

TGERR	In case of reading	In case of writing
0	No toggle error has occurred.	Clears this bit.
1	A toggle error has occurred.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit10]: STUFF (STUFFing error bit)

This bit is a stuffing error flag.

When this bit changes to "1", it indicates that a stuffing error has been detected. If this bit is "0", it indicates that the stuffing error has not been detected. If the stuffing error has been detected, TOUT bit of this register also changes to "1". When "0" is written to this bit, this bit becomes "0". When "1" is written to this bit, the writing is ignored.

STUFF	In case of reading	In case of writing
0	No stuffing error has occurred.	Clears this bit.
1	A stuffing error has occurred.	Ignored.

<Note>

This bit is initialized when the RST bit in the UDC control register (UDCC) is "1".

[bit9, bit8]: HS (Hand Shake status bits)

These bits are hand shake status flags.

These bits indicate the handshake status for transmission/reception.

NULL is indicated when no handshake has occurred caused by an error or when an SOF token is completed with the TKNEN bit in host token endpoint register (HTOKEN).

The bits are updated at the transmission/reception completion time.

bit9	bit8	Handshake Status
0	0	ACK
0	1	NAC
1	0	STALL
1	1	NULL

<Note>

These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".

32.3.4 Host Status Register (HSTATE)

This register indicates the USB circuit status, such as for the connection with a USB device and the transfer mode. Note that the CLKSEL bit is valid in function mode also.

Figure 32.3-4 shows the bit configuration of the host status register (HSTATE).

Figure 32.3-4 Bit configuration of the host status register (HSTATE)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	CLKSEL	SOFBUSY	SUSP	TMODE	CSTAT
Attribute	-	-	R/W	R/W	R/W	R/W	R	R
Initial value	X	X	0	1	0	0	1	0
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	-	×	×	❍	❍	×	×

R/W: Read/Write

R: Read only

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7, bit6]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit5]: Reserved bit

This bit is a reserved bit. Always write "0" to this bit. "0" is read when reading.

[bit4]: CLKSEL (CLocK SELect bit)

This bit specifies the USB operation clock.

Read Value	Explanation
0	The setting is prohibited.
1	A clock for full speed.

<Notes>

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Update this bit only while the RST bit in the UDC control register (UDCC) is "1".
- Always set "1" to this bit. Do not set "0" to this bit.
- Use on-chip bus clock (HCLK) with 13 MHz or over.

[bit3]: SOFBUSY (SOF BUSY bit)

This bit is a SOF busy flag.

This bit changes to "1", when a SOF token is processed in the host token endpoint register (HTOKEN), and it indicates whether the SOF timer is active. If this bit is "0", it indicates that SOF timer is stopped. While SOF timer is operated, SOF timer can be stopped by writing "0" to this bit. When "1" is written to this bit, the writing is ignored.

SOFBUSY	In case of reading	In case of writing
0	The SOF timer is stopped.	Stops the SOF timer.
1	The SOF timer is active.	Ignored.

<Notes>

- This bit is initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- To stop SOF timer, writing "0" to this bit does not stop SOF timer soon. Reading this bit can verify whether it is stopped.

[bit2]: SUSP (SUSPend bit)

This bit is a suspend setting bit.
Writing "1" to the bit sets the suspended state.
If "0" is written to the bit when it is "1" or if the USB bus changes to the k-state, the suspended state is canceled and the RWKIRQ bit in the host interrupt register (HIRQ) changes to "1".

SUSP	Operation
Writing "1"	Sets the suspended state.
Writing "0" when the bit is "1"	Sets the resume state.
Other	The state does not change.

<Notes>

- This bit becomes an initial value when the RST bit in the UDC control register (UDCC) is "1".
- Do not write "1" to this bit during a USB operation (USB bus reset, data transmission/reception during SOF timer operation).
- Do not stop the USB clock in host mode, even in the suspended state.
- If the setting of this bit is changed, the change will not be reflected to the desired state. The change can be verified by reading this bit.

[bit1]: TMODE (Transmission MODE bit)

This bit is a transfer mode flag.

When this bit is "1", it indicates that the full speed device is connected. When this bit is "0", it indicates that the low speed device is connected. It is valid when the CSTAT bit in host status register (HSTATE) is "1".

Read Value	Explanation
0	Low speed (This series does not support the low speed.)
1	Full speed

<Notes>

- This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).
- Use on-chip bus clock (HCLK) with 13MHz or over.
- This series does not support the low speed.

[bit0]: CSTAT (Connect STATus bit)

This bit is a connection status flag.

When this bit is "1", it indicates that the device is connected. When this bit is "0", it indicates that the device is disconnected.

Read Value	Explanation
0	The USB device is disconnected.
1	The USB device is connected.

<Note>

This bit is not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

32.3.5 SOF Interrupt FRAME comparison register (HFCOMP)

This register specifies the data that is compared with the lower 8-bit of the frame number for SOF token processing. The register is compared with the lower 8-bit of the frame number, when the SOFSTEP bit in the host control register 0 (HCNT0) is "0". If the SOFIRE bit is "1", that is, the comparison result is a match, the SOFIRQ bit in the host interrupt register (HIRQ) changes to "1" at the start of SOF transmission, generating an interrupt request.

Figure 32.3-5 shows the bit configuration of the SOF interrupt FRAME comparison register (HFCOMP).

Figure 32.3-5 Bit configuration of the SOF interrupt FRAME comparison register (HFCOMP)

bit	15	14	13	12	11	10	9	8
	FRAMECOMP							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15 to bit8]: FRAMECOMP

These bits are FRAME comparison data.

This register specifies the data that is compared with the lower 8-bit of the frame number for SOF token processing.

When the SOFSTEP bit in the host control register 0 (HCNT0) is "0", this register is compared with the frame number for SOF token processing. If the comparison result is a match, "1" is set to the SOFIRQ bit of host interrupt register (HIRQ).

When the SOFSTEP bit in the host control register 0 (HCNT0) is "0", the setting of this register is invalid.

<Note>

These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

32.3.6 Retry Timer Setting Register (HRTIMER)

This register sets the retry interval for tokens.

Figure 32.3-6 shows the bit configuration of the retry timer setting register (HRTIMER).

Figure 32.3-6 Bit configuration of the retry timer setting register (HRTIMER)

Retry Timer Setting Register (HRTIMER)

bit	7(23)	6(22)	5(21)	4(20)	3(19)	2(18)	1(17)	0(16)
	Reserved						RTIMER2	
Attribute	-						R/W	
Initial value	X						00	
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-						×	

bit	15	14	13	12	11	10	9	8
	RTIMER1							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

bit	7	6	5	4	3	2	1	0
	RTIMER0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit23 to bit18]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit17 to bit0]: RTIMER2 to RTIMER0

These bits set the retry interval for tokens. The retry timer is activated by the start of processing of a token, decrementing the timer by 1 at the 1-bit transfer clock (12 MHz at full speed). Once the retry timer reaches 0, it will be completed after the token is processed. If a retry is made for a token in the EOF area, the retry timer stops until SOF processing is completed. The value of the stopped timer is decremented by 1 after the SOF processing is completed.

<Note>

These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

32.3.7 Host Address Register (HADR)

This register is used for the address field at the token transmission time.

Figure 32.3-7 shows the bit configuration of the host address register (HADR).

Figure 32.3-7 Bit configuration of the host address register (HADR)

bit	15	14	13	12	11	10	9	8
	Reserved	Address						
Attribute	-	R/W						
Initial value	X	0000000						
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	×						

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15]: Reserved bit

This bit is a reserved bit. It is undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit14 to bit8]: Address

These bits are address bits.

These bits specify the token address.

<Note>

These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

32.3.8 EOF Setting Register (HEOF)

This register sets the period in which token processing is disabled prior to SOF token processing. In a comparison of the SOF timer value and the data in this register, if the SOF timer value is smaller than this data and a processing request is issued for an IN token, OUT token, or SETUP token, the requested token is processed after the processing of the SOF token. This is done to avoid concurrent processing of SOF tokens generated by the hardware and other tokens. This register has the time unit of 1-bit transfer time.

Figure 32.3-8 shows the bit configuration of the EOF setting register (HEOF).

Figure 32.3-8 Bit configuration of the EOF setting register (HEOF)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	EOF1					
Attribute	-	-	R/W					
Initial value	X	X	000000					
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	-	×					

bit	7							0
	EOF0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	×							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15, bit14]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit13 to bit0]: EOF1, EOF0 (End Of Frame bits)

These bits are EOF bits.

These registers set the period in which token processing is disabled prior to SOF token processing. Set these registers with a margin with 1 packet length or longer. These registers have the time unit of 1-bit transfer time.

For example, suppose that MAXPKT = 64 bytes at full speed, then a value of "$2C9_H$" will be set in these bits according to the following formula:

(Token_length + packet_length + header + CRC)*7/6 + Turn_around_time

= (34 bits + 546 bits)*7/6 + 36 bits

= 712.7 bits

<Note>

These bits are not initialized by setting "1" to the RST bit in the UDC control register (UDCC).

32.3.9 FRAME Setting Register (HFRAME)

This register sets the frame number for SOF token processing. If SOF activation is set in the TKNEN bit in the host token endpoint register (HTOKEN), the SOF timer is activated and then SOF is automatically transmitted every 1 ms. The register is automatically incremented by 1 every time that SOF processing is completed.

Figure 32.3-9 shows the bit configuration of the FRAME setting register (HFRAME).

Figure 32.3-9 Bit configuration of the FRAME setting register (HFRAME)

bit	15	14	13	12	11	10	9	8
	Reserved	Reserved	Reserved	Reserved	Reserved	FRAME1		
Attribute	-	-	-	-	-	R/W		
Initial value	X	X	X	X	X	000		
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	-	-	-	-	-	❍		

bit	7	6	5	4	3	2	1	0
	FRAME0							
Attribute	R/W							
Initial value	00000000							
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍							

R/W: Read/Write

-: Undefined

X: Undefined

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit15 to bit11]: Reserved bits

These bits are reserved bits. They are undefined when reading. The operation of LSI will not be affected either "0" or "1" is written.

[bit10 to bit0]: FRAME1, FRAME0

These bits are frame setting bits.

They specify the number of frames of SOF.

<Notes>

- These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".
- Before setting SOF to the TKNEN bit in the host token endpoint register (HTOKEN), specify the number of frames in this register.
- Do not write to these bits while SOF token processing is in progress and the SOFBUSY bit in the host status register (HSTATE) is "1".

32.3.10 Host Token Endpoint Register (HTOKEN)

This register sets the toggle, an endpoint, and tokens.

Figure 32.3-10 shows the bit configuration of the host token endpoint register (HTOKEN).

Figure 32.3-10 Bit configuration of the host token endpoint register (HTOKEN)

bit	7	6	5	4	3	2	1	0
	TGGL	TKNEN			ENDPT			
Attribute	R/W	R/W			R/W			
Initial value	0	000			0000			
Initialized(❍)/Not-initialized(×) with RST bit in UDCC	❍	❍			❍			

R/W: Read/Write

Initialized(❍)/Not-initialized(×) with RST bit in UDCC

[bit7]: TGGL (ToGGLe bit)

This bit is a toggle bit.

This bit specifies the data toggle. In transmission, toggle data is transmitted according to the setting of this bit. In reception, received toggle data and the toggle data indicated by this bit are compared to detect any errors.

Written value	Explanation
0	DATA0
1	DATA1

<Notes>

- This bit becomes an initial value when the RST bit in the UDC control register (UDCC) is "1".
- Set this bit when the TKNEN bit in host token endpoint register (HTOKEN) is "000_B".

[bit6 to bit4]: TKNEN (ToKeN ENable bits)

These bit are token enable bits.

Tokens are transmitted according to the setting of these bits. The bits change to "000_B" after the operation is completed, and the CMPIRQ bit in the host interrupt register (HIRQ) changes to "1". If the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated.

The settings of the TGGL and ENDPT bits in SOF token processing are ignored.

bit6	bit5	bit4	Operation
0	0	0	The token is not transmitted.
0	0	1	A SETUP token is transmitted.
0	1	0	An IN token is transmitted.
0	1	1	An OUT token is transmitted.
1	0	0	An SOF token is transmitted.
1	0	1	Reserved (The setting is prohibited.)
1	1	0	Reserved (The setting is prohibited.)
1	1	1	Reserved (The setting is prohibited.)

<Notes>

- These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".
- PRE packets are not supported.
- Do not set these bits to "100_B" when the SOFBUSY bit in the host status register (HSTATE) is "1".
- Set to host mode before writing to these bits.
- In cases of issuing the token again after an interrupt flag (CMPIRQ) becomes "1", write to these bits after waiting for 3 or more cycles of the USB transfer clock (12 MHz at full speed).
- In the disconnected state (CSTAT="0" in HSTATE), the token is not processed even if the bits are written.

[bit3 to bit0]: ENDPT (ENDPoinT bits)

These bits are endpoint bits.

These bits set the endpoint used to transmit/receive data to/from a USB device.

<Note>

These bits become initial values when the RST bit in the UDC control register (UDCC) is "1".

32.4 Explanation of Operations and Setting Procedure Examples

This section explains operations of USB Mini-HOST. Also, examples of procedures for setting the operating state are shown.

32.4.1 Connecting a USB Device

This section explains how to detect the connections with external USB devices through software.

■ Setting the Mini-HOST function

To allow the USB Mini-HOST to operate, disable USB operation (USBEN = 0) in the USB enable register (USBEN), enable USB clock output in the USB clock generating part setting, and then enable USB operation (USBEN = 1).

Then, write "1" to the HOST bit in host control register 0 (HCNT0) for operation as a host.

■ Disconnected/Connected state of an external USB device

When an external USB device is disconnected, the USB Mini-HOST UDP/UDM pins are at the "L" level as determined through a pull-down resistor. At this time, the CSTAT bit in the host status register (HSTATE) is "0", and the TMODE bit is undefined. If the external USB device is connected again, the CSTAT bit in the host status register (HSTATE) changes to "1".

■ Detecting the connections with external USB devices

When detecting the connections with external USB devices, the CNNIRQ bit in the host interrupt register (HIRQ) changes to "1", and the CNNIRE bit in the host control register 0 (HCNT0) is set to "1", and a device connection interrupt request is generated. To clear this interrupt request, write "0" to the CNNIRQ bit in the host interrupt register (HIRQ). To detect the connections with USB devices through polling instead of interrupts, write "0" to the CNNIRE bit in host control register 0 (HCNT0), and write a program for confirming that the CNNIRQ bit in the host interrupt register (HIRQ) changes to "1".

■ Obtaining a transfer speed and selecting the clock for a connected USB device

To obtain an available transfer speed for a connected USB device after its connection was detected, see the value of the TMODE bit in the host status register (HSTATE). Table 32.4-1 shows the relationship between transfer speeds and the TMODE bit in the host status register (HSTATE).

Table 32.4-1 Relationship between transfer speeds for a connected USB device and TMODE

Transfer Speed for Connected USB Device	TMODE
Full speed	1
Low speed	0

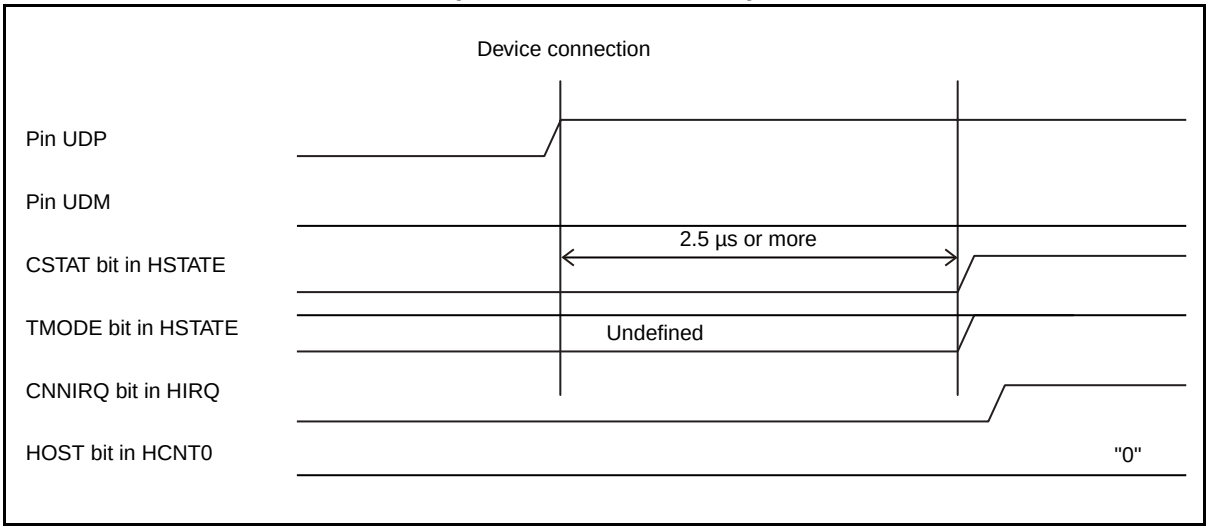
<Note>

This series does not support the low speed.

After obtaining the transfer speed for the external USB device, if the RST bit in the UDC control register (UDCC) is "1", update the CLKSEL bit in the host status register (HSTATE) according to the obtained transfer speed.

Figure 32.4-1 shows an example of the timing in detection of the connection with a full-speed device.

Figure 32.4-1 Example of the timing in detection of the connection with a full-speed device (HOST bit in HCNT0 = 0)



<Notes>

- The CSTAT bit in the host status register (HSTATE) changes to "1" after 2.5 µs elapses after the external USB device is connected.
- The TMODE bit and CSTAT bit in the host status register (HSTATE) are updated irrespective of the value of the HOST bit in host control register 0 (HCNT0). CNNIRQ bit and DIRQ bit of host interrupt register (HIRQ) become "1" if the conditions are satisfied.

32.4.2 Resetting the USB Bus

This section explains how to reset the USB bus in host mode.

■ Resetting the USB bus

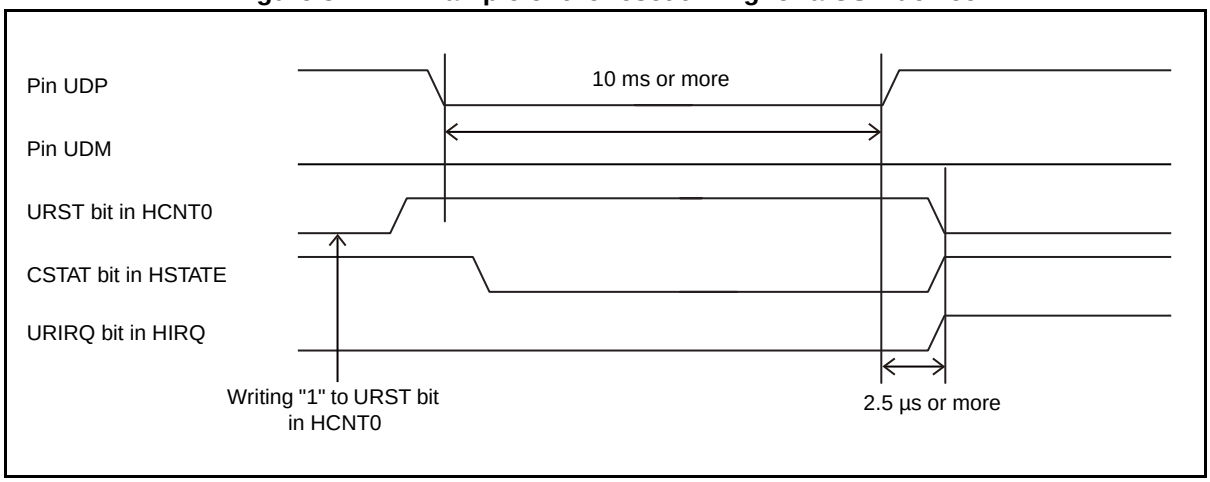
Writing "1" to the URST bit in host control register 0 (HCNT0) in host mode results in SE0 being transmitted for 10 ms or more to reset the USB bus. The URST bit in host control register 0 (HCNT0) changes to "0" at the end of the USB bus reset. At this point, if the URIRE bit in host control register 0 (HCNT0) is "1", an interrupt request is generated, and the URIRQ bit in the host interrupt register (HIRQ) changes to "1". To clear the interrupt request, write "0" to the URIRQ bit in the host interrupt register (HIRQ).

Note the following points about resetting the USB bus:

1. Before resetting the USB bus, confirm that the CSTAT bit in the host status register (HSTATE) is "1" and the USB device is connected.
2. When the USB bus is reset, the CSTAT bit in the host status register (HSTATE) changes to "0", indicating the disconnected state. The DIRQ bit in the host interrupt register (HIRQ) does not change to "1" at this time.
3. After resetting the USB bus, compare the CLKSEL bit and TMODE bit in the host status register (HSTATE). If they do not match, update the CLKSEL bit so that they match. Update this bit only while the RST bit in the UDC control register (UDCC) is "1".

Figure 32.4-2 shows an example of the reset timing for a USB device.

Figure 32.4-2 Example of the reset timing for a USB device



32.4.3 Token Packets

This section explains how to set token packets.
The setting is unnecessary if there is no change in each register (HADR, EP1C, EP2C, HFRAME and HEOF).

■ How to set token packets

In host mode, the buffer of endpoint 1 and the buffer of endpoint 2 are used as the buffer for transmission/reception. In case of processing either an IN token, OUT token, or SETUP token in host mode, specify the destination addresses in the host address register (HADR). Also, in the PKS bit and DIR bit in the EP1 control register (EP1C) or EP2 control register (EP2C), specify the maximum number of bytes per packet and the transfer direction.

If the DIR bit in the EP1 control register (EP1C) is "1", the buffer of endpoint 1 is used as an OUT-direction buffer, and the buffer of endpoint 2 is used as an IN-direction buffer. In this case, write "0" to the DIR bit in the EP2 control register (EP2C).

If the DIR bit in the EP1 control register (EP1C) is "0", the buffer of endpoint 1 is used as an IN-direction buffer, and the buffer of endpoint 2 is used as an OUT-direction buffer. In this case, write "1" to the DIR bit in the EP2 control register (EP2C).

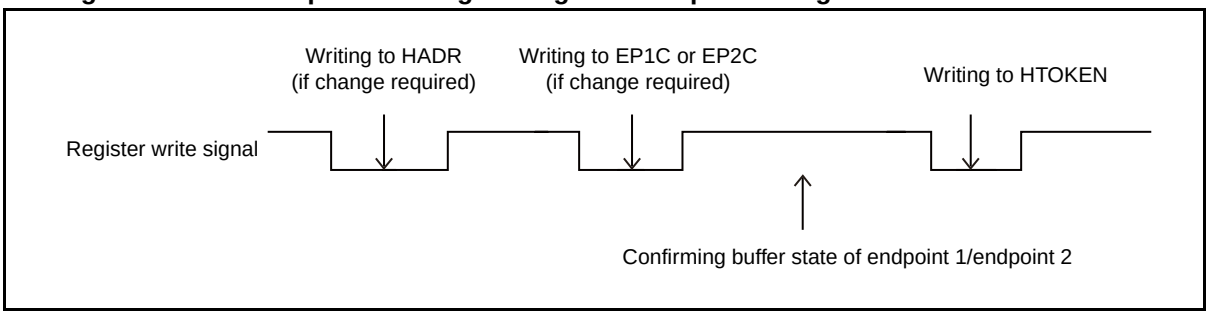
To process the token, follow the following procedures for the settings.

1) Set the DIR bit and PKS bit in the EP1 control register (EP1C) and EP2 control register (EP2C).
2) If the endpoint n (n=1 or 2) to be used is used in an OUT-direction, write the transmitted data in the buffer of the endpoint n (n=1 or 2), and set "0" to the DRQ bit in the EPn status register (EpnS:n=1 or 2). If the direction is in IN-direction, read the DRQ bit of the EPn status register (EPnS:n=1 or 2), and confirm that "0" is written.
3) Set the target endpoint, token and toggle data in the host token endpoint register (HTOKEN).

The USB circuit transmits token packets in the order of Sync, token, address, endpoint, CRC5, and EOP according to the specified tokens. (Sync, CRC5, and EOP are automatically transmitted.) After the end of each packet, the CMPIRQ bit in the host interrupt register (HIRQ) changes to "1", and TKNEN in the host token endpoint register (HTOKEN) becomes "000_B". At this point, if the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. To clear the interrupt request, write "0" to the CMPIRQ bit in the host interrupt register (HIRQ).

Figure 32.4-3 shows an example of setting the registers for processing of IN/OUT/SETUP tokens.

Figure 32.4-3 Example of setting the registers for processing of IN/OUT/SETUP tokens



For an SOF token, writing an EOF time and frame number to the EOF setting register (HEOF) and FRAME setting register (HFRAME), respectively, and writing the SOF token code to the TKNEN bit in the host token endpoint register (HTOKEN) results in Sync, the SOF token, the frame number, CRC5, and EOP being transmitted. At this time, the SOFBUSY bit in the host status register (HSTATE) changes to "1", and the FRAME setting register (HFRAME) is incremented by 1. Also at this time, CMPIRQ in the host interrupt register (HIRQ) changes to "1", and the TKNEN bit in the host token endpoint register (HTOKEN) clears to "000_B". At this point, if the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. Subsequently, interrupt requests by CMPIRQ will not be generated for automatically generated SOF. To clear the interrupt request by the end of the token, write "0" to the CMPIRQ bit in the host interrupt register (HIRQ).

While the SOFBUSY bit in the host status register (HSTATE) is "1", SOF is automatically transmitted every 1 ms. The SOFBUSY bit in the host status register (HSTATE) changes to "0" under the following conditions (conditions to stop SOF):

- "0" is written to the SOFBUSY bit in the host status register (HSTATE).
- The USB bus is reset (by writing "1" to the URST bit in host control register 0 (HCNT0)).
- "1" is written to the SUSP bit in the host status register (HSTATE).
- The USB device is disconnected (the CSTAT bit in HSTATE is "0").

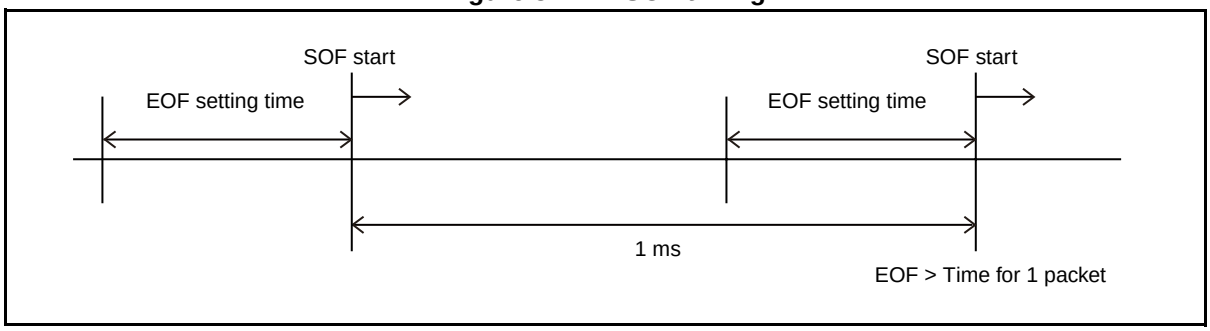
Before switching from host mode to function mode, be sure to write "0" to the SOFBUSY bit in the host status register (HSTATE) and confirm that the SOFBUSY bit has changed to "0" and the TKNEN bit of host token endpoint register (HTOKEN) is "000_B" and confirm that the SUSP bit in host status register (HSTATE) is "0" and set the RST bit in UDC control register (UDCC) to "1".

To change the SOFBUSY bit in the host status register (HSTATE) back to "1", process the SOF token again.

If the TKNEN bit in the host token endpoint register (HTOKEN) is written between the EOF setting time and SOF start time, the EOF setting register (HEOF) will wait until SOF is processed before processing the specified token in order to avoid concurrent processing of the SOF token and other tokens. The EOF setting register (HEOF) has a 1-bit unit of time. For example, suppose that "10_H" is written to EOF bit in the EOF setting register (HEOF), then the value will be 16*1/12 MHz = 1333.3 ns in full speed mode. If the specified EOF setting time is shorter than the time for one packet, SOF processing may overlap with processing of another token. In such cases, the LSTSOF bit in the host error status register (HERR) changes to "1", and SOF will not be processed. If LSTSOF in the host error status register (HERR) changes to "1", increase the time in the data for the EOF setting register. (See "32.3.8 EOF Setting Register (HEOF)".)

Figure 32.4-4 shows the SOF timing.

Figure 32.4-4 SOF timing



32.4.4 Data Packets

This section explains how to transfer data packets.

A data packet is transferred in the following procedure after a token packet is transmitted.

- In transmission
  1. Sync is automatically transmitted.
  2. If the TGGL bit in the host token endpoint register (HTOKEN) is "0", DATA0 is transmitted; if it is "1", DATA1 is transmitted.
  3. To transmit all transmitted data, select the buffer of endpoint 1 if the DIR bit in the EP1 control register (EP1C) is "1", or select the buffer of endpoint 2 if the DIR bit is "0".
  4. 16-bit CRC is transmitted.
  5. 2-bit EOP is transmitted.
  6. 1-bit J State is transmitted.

- In reception
  1. Receive Sync.
  2. Toggle data is received and compared with the TGGL bit in the host token endpoint register (HTOKEN).
  3. To sort the received data when the toggle data and the TGGL bit are found to match, select the buffer of endpoint 2 if the DIR bit in the EP1 control register (EP1C) is "1", or select the buffer of endpoint 1 if the DIR bit is "0".
  4. 16-bit CRC is verified when EOF is received.

If the HOST bit in host control register 0 (HCNT0) is "1", write inverted values as the data in the DIR bits in the EP1 control register (EP1C) and EP2 control register (EP2C). For example, if the DIR bit in the EP1 control register (EP1C) is "0", write "1" to the DIR bit in the EP2 control register (EP2C).

32.4.5 Operation of a Handshake Packet

This section explains the operation of a handshake packet that notifies the transmitter/receiver of the host state.

The handshake packet determines whether the receiving side is in a state where it is capable of properly receiving data and the receiving side transmits ACK, NAK, or STALL. If the USB circuit receives the handshake packet, the type of the received handshake packet is written to the HS bit in the host error status register (HERR). If the handshake packet is transmitted, the type of the transmitted handshake packet is written to the HS bit in the host error status register (HERR).

32.4.6 Retry Function

If an error such as the NAK error or CRC error occurs at the packet end time and the RETRY bit in host control register 1 (HCNT1) is "1", the operation is retried for the period specified in the retry timer register (HRTIMER).
This section explains operations of the retry function.

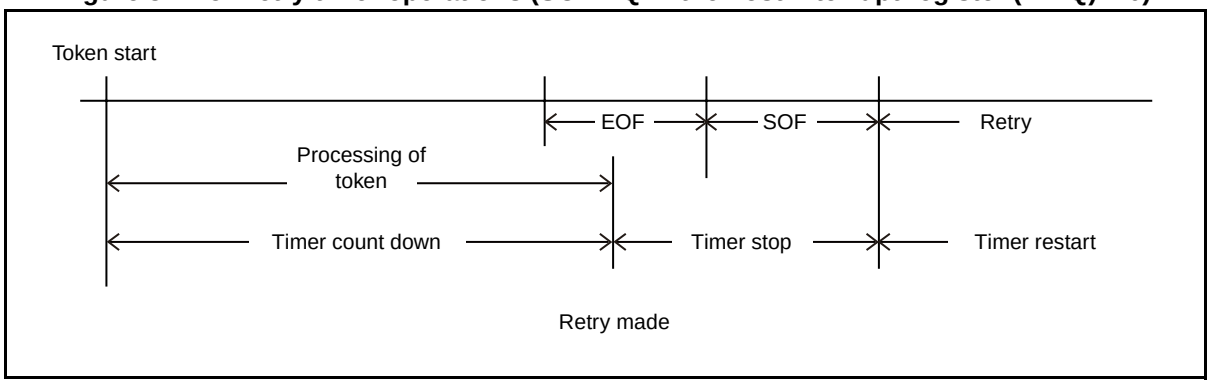
If an error other than the STALL error or device disconnection error (the HS bit of the host error status register (HERR) is "01_B", RERR bit is "1", TOUT bit is "1", CRC bit is "1", or STUFF bit is "1") occurs during transfer and the RETRY bit in host control register 1 (HCNT1) is "1", the operation for the token involved is retried. The retries are ended under the following conditions:

- "0" is written to the RETRY bit in host control register 1 (HCNT1).
- A retry timer value of 0 is detected.
- The SOFIRQ bit in host interrupt register (HIRQ) becomes "1" by SOF.
- ACK is detected.
- The disconnection of the device is detected.

The retry timer is activated at the start of token transfer, and it counts down using a 1-bit transfer clock. However, if a retry is made in the EOF area, the counting is stopped. If the SOF token ends with "0" in the SOFIRQ bit in the host interrupt register (HIRQ), the timer resumes from the "stopped" timer value. At the point where the retry timer changes to "0", the packet transfer ends and the CMPIRQ bit of the host interrupt register (HIRQ) is set to "1".

Figure 32.4-5 shows operations of the retry timer with "0" in the SOFIRQ bit in the host interrupt register (HIRQ).

Figure 32.4-5 Retry timer operations (SOFIRQ in the host interrupt register (HIRQ) = 0)



Completion information for the completed packet is written to each register when the retry operation is completed.

32.4.7 SOF Interrupts

This section explains operations of SOF interrupts.

If the SOFSTEP bit of the host control register 1 (HCNT1) is "0", the SOF interrupt FRAME comparison register (HFCOMP) is compared with the lower 8-bit of the frame number for SOF token, if the comparison result is a match, the SOFIRQ bit of the host interrupt register (HIRQ) is set to "1" at transmission of SOF.

If the SOFSTEP bit of the host control register 1 (HCNT1) is "1", SOFIRQ bit of host interrupt register (HIRQ) is set to "1" every time that SOF is transmitted.

If the SOFIRE bit of host control register 0 (HONT0) is "1", and changing the SOFIRQ bit of host interrupt register (HIRQ) to "1" generates an interrupt. SOF processing by the host token end point register (HTOKEN) does not change the SOFIRQ bit of the host interrupt register (HIRQ) to "1".

Figure 32.4-6 and Figure 32.4-7 show examples of operations of SOF interrupts.

Figure 32.4-6 Example of SOF interrupt operations (SOFSTEP in host control register 1 (HCNT1) = 1)

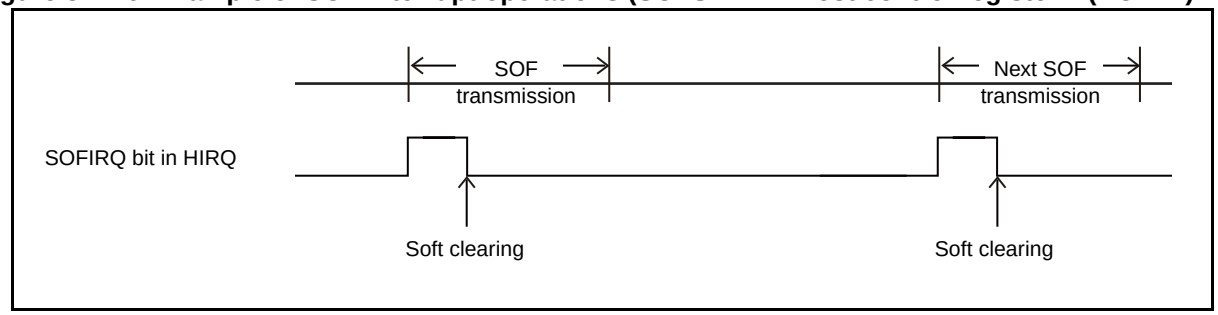
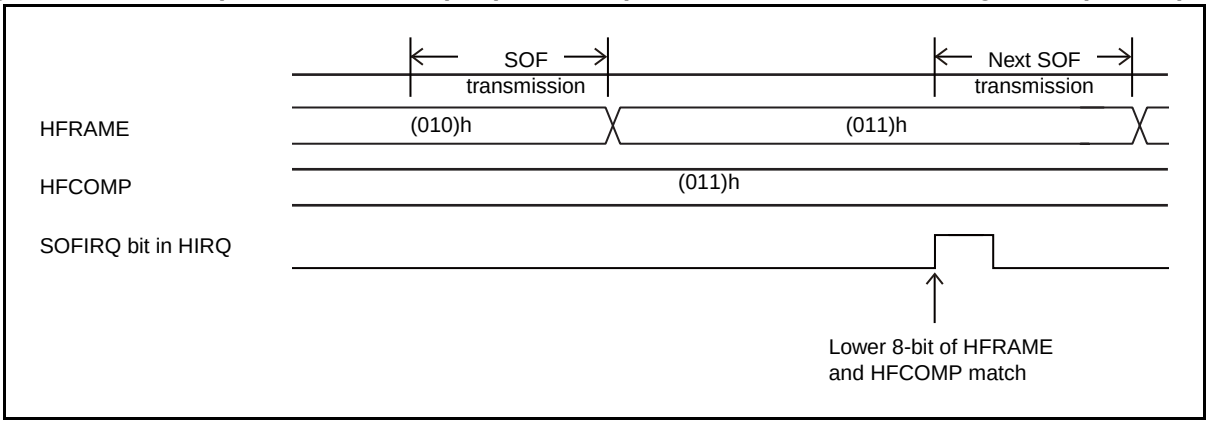


Figure 32.4-7 Example of SOF interrupt operations (SOFSTEP in host control register 1 (HCNT1) = 0)



If "1" is written to the CANCEL bit in host control register 1 (HCNT1), a token other than an SOF token is set in the host token endpoint register (HTOKEN) in the EOF area, and when the SOFIRQ bit in the host interrupt register (HIRQ) changes to "1" at the next SOF transmission, that token will not be processed and the TKNEN bit in the host token endpoint register (HTOKEN) will be cleared to "000_B".

In this case, the CMPIRQ bit in the host interrupt register (HIRQ) does not change to "1". Cancellation of the token can be known from the TCAN bit in the host interrupt register (HIRQ) when the SOFIRQ bit changes to "1". To process the token again, write "0" to the TCAN bit in the host interrupt register (HIRQ), and write the token to the TKNEN bit in the host token endpoint register (HTOKEN).

If "0" is written to the CANCEL bit in host control register 1 (HCNT1), the token specified in the host token endpoint register (HTOKEN) is processed after SOF transmission.

Figure 32.4-8 shows an example of token cancel operations, and Figure 32.4-9 shows an example of token operations.

Figure 32.4-8 Example of token cancel operations (CANCEL in host control register 1 (HCNT1) = 1)

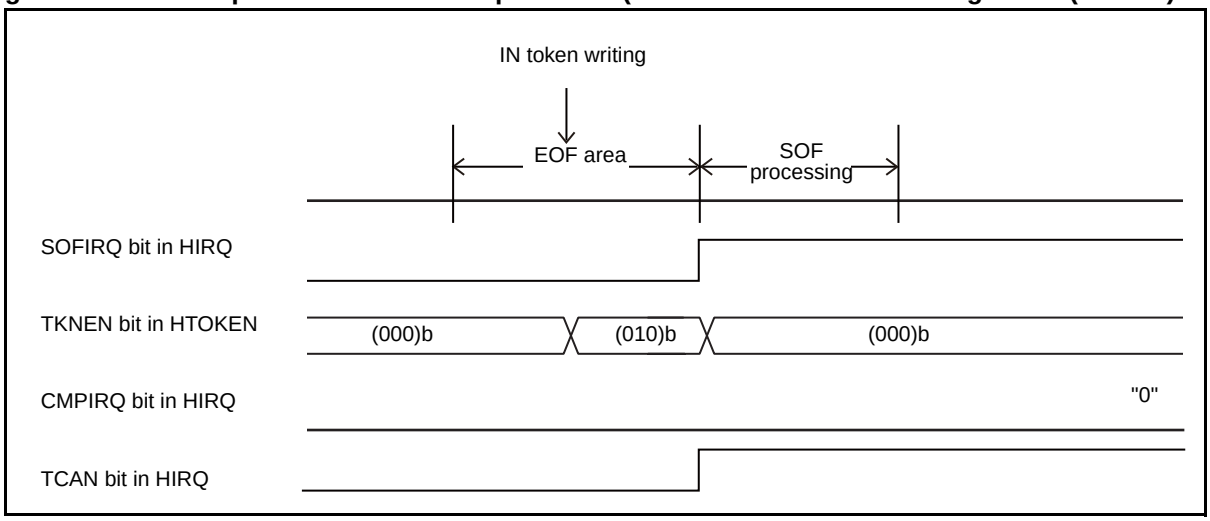
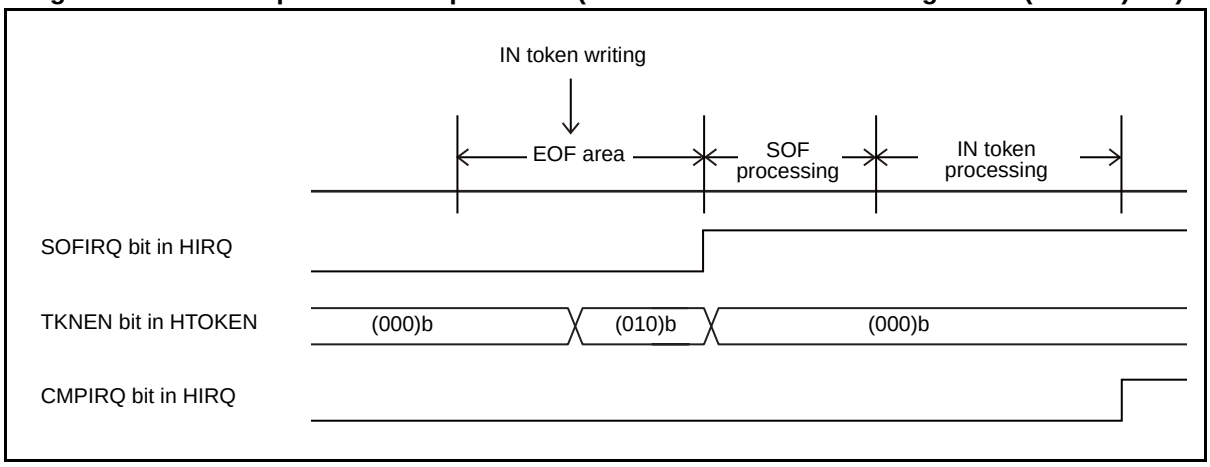


Figure 32.4-9 Example of token operations (CANCEL in host control register 1 (HCNT1) = 0)



32.4.8 Error Status

USB Mini-HOST supports various error information.
This section explains error statuses.

■ Stuffing error

When 6 consecutive "1" bits are detected, 1-bit of "0" should be inserted. However, when 7 consecutive "1" bits are detected, a stuffing error occurs and the STUFF bit in the host error status register (HERR) changes to "1". To clear the STUFF bit, write "0" to it. If the next token is processed without clearing the STUFF bit, the bit is updated at the time that the next token is completed.

■ Toggle error

The toggle data for a data packet and the TGGL bit in the host token endpoint register (HTOKEN) are compared for an IN token, and if they do not match, a toggle error occurs and the TGERR bit in the host error register (HERR) changes to "1". To clear the TGERR bit in the host error register (HERR), write "0" to the TGERR bit. If the next token is processed without clearing the TGERR bit, the bit is updated at the time that the next token is completed.

■ CRC error

Data for a received data packet and CRC for an IN token are calculated using a polynomial expression for CRC, "$G(X)=X^{16}+X^{15}+X^{2}+1$", and a remainder other than "$800d_H$" indicates that a CRC error has occurred, which causes the CRC bit in the host error register (HERR) to change to "1". To clear the CRC bit in the host error register (HERR), write "0" to the CRC bit. If the next token is processed without clearing the CRC bit, the bit is updated at the time that the next token is completed.

■ Time out error

When a data packet or handshake has not been input within the given period, SE0 is detected in received data, or a stuffing error is detected, TOUT bit in the host error status register (HERR) changes to "1". To clear the TOUT bit in the host error register (HERR), write "0" to the TOUT bit. If the next token is processed without clearing the TOUT bit, the bit is updated at the time that the next token is completed.

■ Receive error

The received packet size is the value in the PKS bit in the EP1 control register (EP1C) when EP1 is used as the receive buffer, and the value in the PKS bit in the EP2 control register (EP2C) when EP2 is used as the receive buffer. If more data is received than the received packet size, the RERR bit in the host error status register (HERR) changes to "1". To clear the RERR bit in the host error register (HERR), write "0" to the RERR bit. If the next token is processed without clearing the RERR bit, the bit is updated at the time that the next token is completed.

32.4.9 Packet End

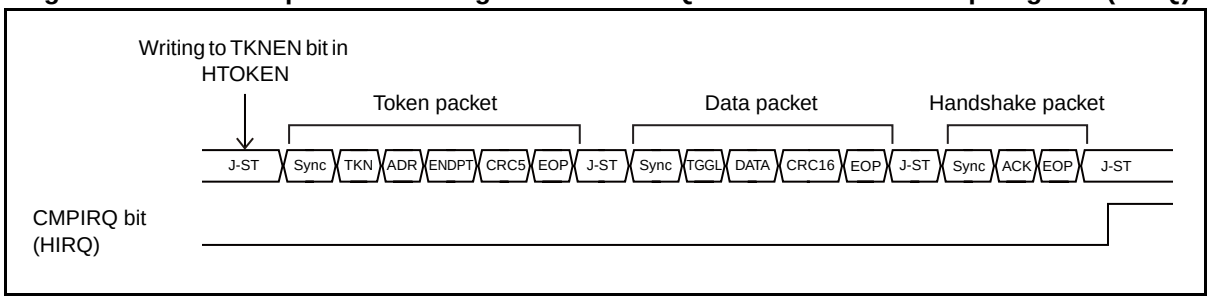
This section explains the operations at the packet end time.

The CMPIRQ bit in the host interrupt register (HIRQ) changes to "1" at the end of a packet in USB Mini-HOST. At this point, when the CMPIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated.

An interrupt flag is generated at the end of a packet.

Figure 32.4-10 shows an example of the packet end timing when the TKNEN bit in the host token endpoint register (HTOKEN) is "001_B", "010_B", or "011_B" (SETUP token, IN token, or OUT token).

Figure 32.4-10 Example 1: Set timing for the CMPIRQ bit in the host interrupt register (HIRQ)



J-ST: J State

TKN: Token

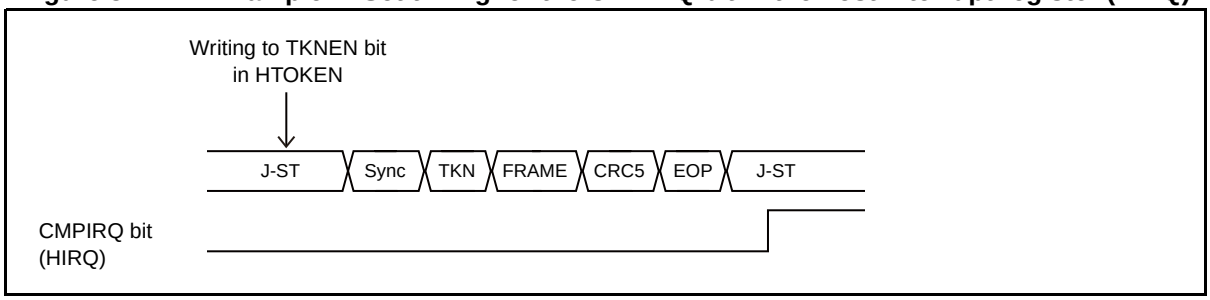
ADR: Address

ENDPT: Endpoint

TGGL: Toggle

Figure 32.4-11 shows an example of the packet end timing when the TKNEN bit in the host token endpoint register (HTOKEN) is "100_B" (SOF token).

Figure 32.4-11 Example 2: Set timing for the CMPIRQ bit in the host interrupt register (HIRQ)



J-ST: J State

TKN: Token

FRAME: Frame number

32.4.10 Suspend and Resume

USB Mini-HOST supports suspend and resume.
This section explains the suspend and resume operations.

■ Suspend operation

Writing "1" to the SUSP bit in the host status register (HSTATE) places the USB bus in a high impedance state, stopping the circuit blocks that do not require a clock and suspending the USB circuit. The SUSP bit in the host status register (HSTATE) changes to "1" when the USB circuit enters the suspended state.

Do not specify suspend during a USB bus reset when the SOFBUSY bit in the host status register (HSTATE) is "1" and during data transmission or reception. Also, do not stop the clock feed to USB in the suspended state. The following operations can stop the clock:

- Transition to stop mode or watch mode
- Writing of "0" to the UCEN bit in the USB clock configuration register (UCCR)

■ Resume operation

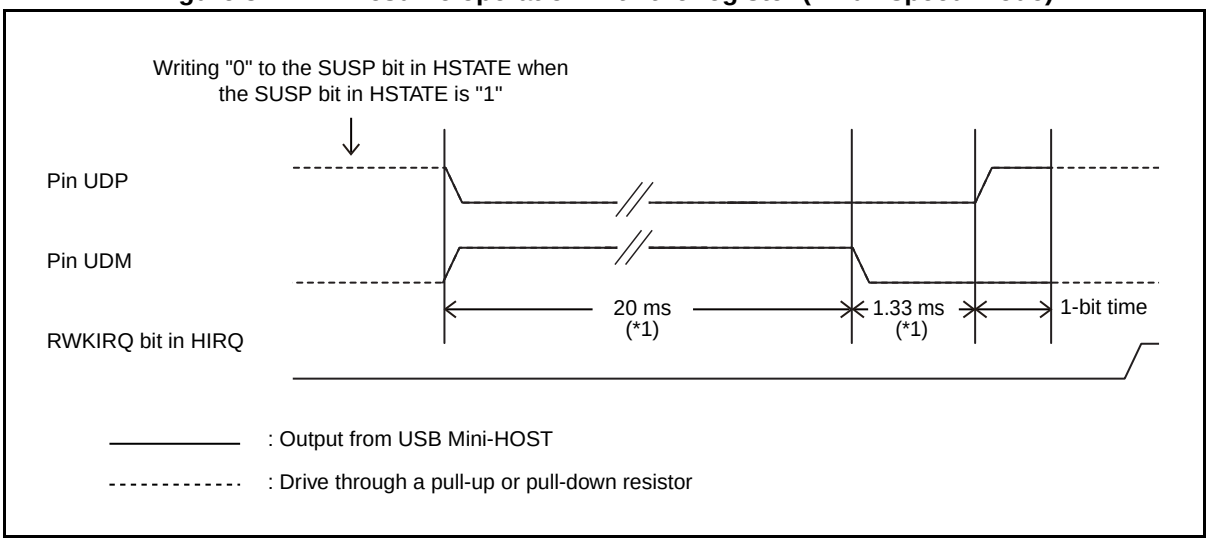
If any of the following conditions are satisfied, the resume operation from the suspended state is started:

1. "0" is written to the SUSP bit in the host status register (HSTATE).
2. UDP/UDM pins in the K State are detected.
3. The disconnection of the device is detected.
4. The connection of the device is detected.

If the RWKIRQ bit in the host interrupt register (HIRQ) changes to "1", tokens can be issued.

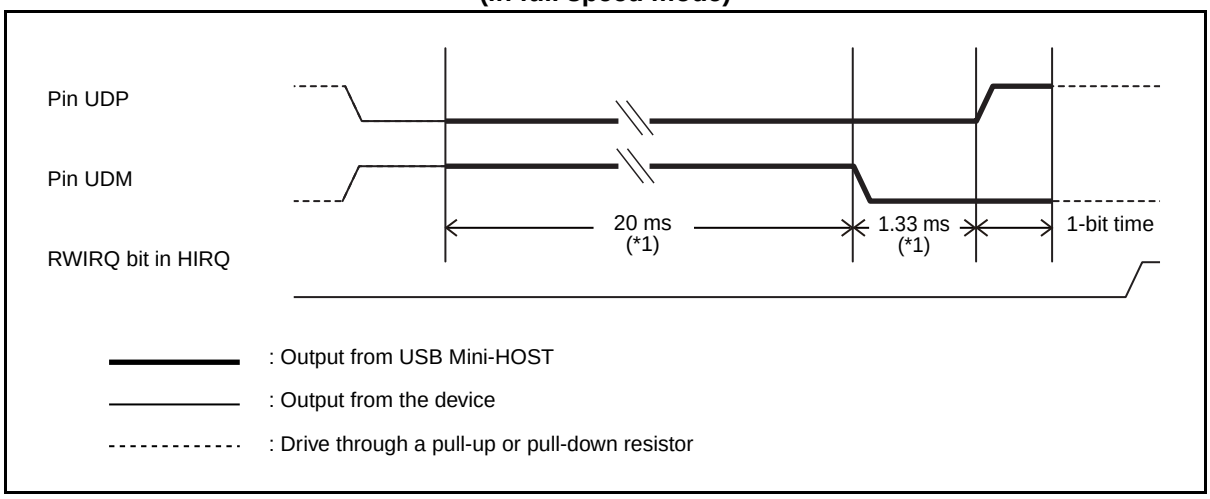
Figure 32.4-12, Figure 32.4-13, Figure 32.4-14, and Figure 32.4-15 show the resume operations caused by conditions 1, 2, 3, and 4 respectively.

Figure 32.4-12 Resume operation with the register (in full speed mode)



*1 The value is not guaranteed.

Figure 32.4-13 Resume operation from a detection of K state by UDP/UDM pins (in full speed mode)



*1 The value is not guaranteed.

Figure 32.4-14 Resume operation for a disconnected device

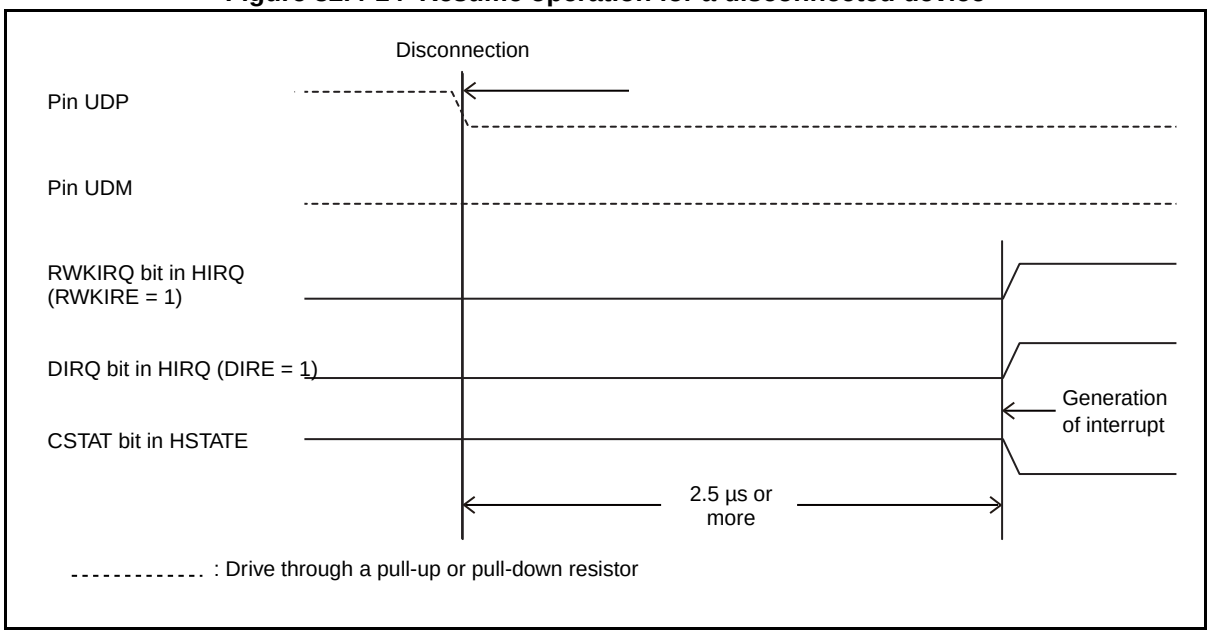
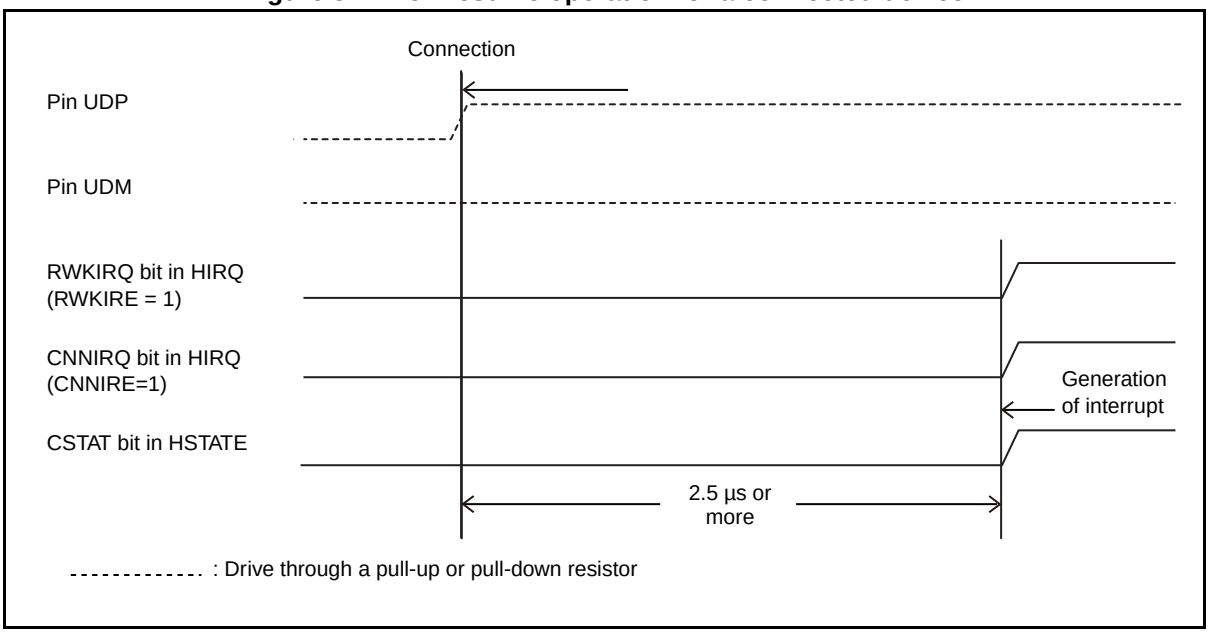


Figure 32.4-15 Resume operation for a connected device



32.4.11 Disconnecting a USB Device

This section explains the operation of disconnecting a USB device.

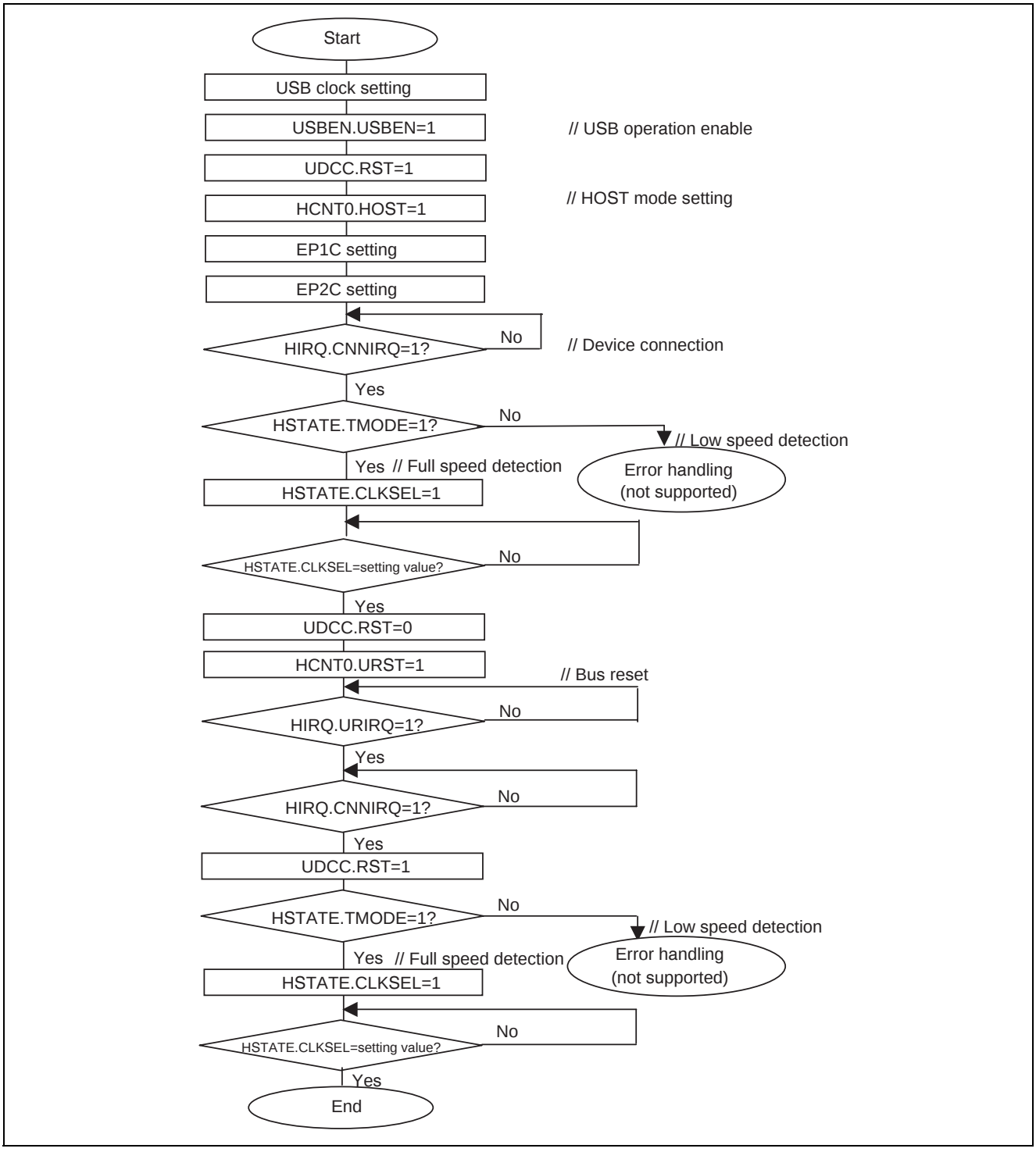
If both the UDP/UDM pins are at the "L" level for 2.5 µs or more, the USB device is judged disconnected, causing the CSTAT bit in the host status register (HSTATE) to change to "0", and the DIRQ bit in host interrupt register (HIRQ) changes to "1". If the DIRE bit in the host control register 0 (HCNT0) is "1", an interrupt request is generated. To clear this interrupt request, write "0" to the DIRQ bit in the host interrupt register (HIRQ).

When the USB bus is reset, the USB device is judged disconnected, which causes the CSTAT bit in the host status register (HSTATE) to change to "0", but the DIRQ bit in the host interrupt register (HIRQ) does not change to "1".

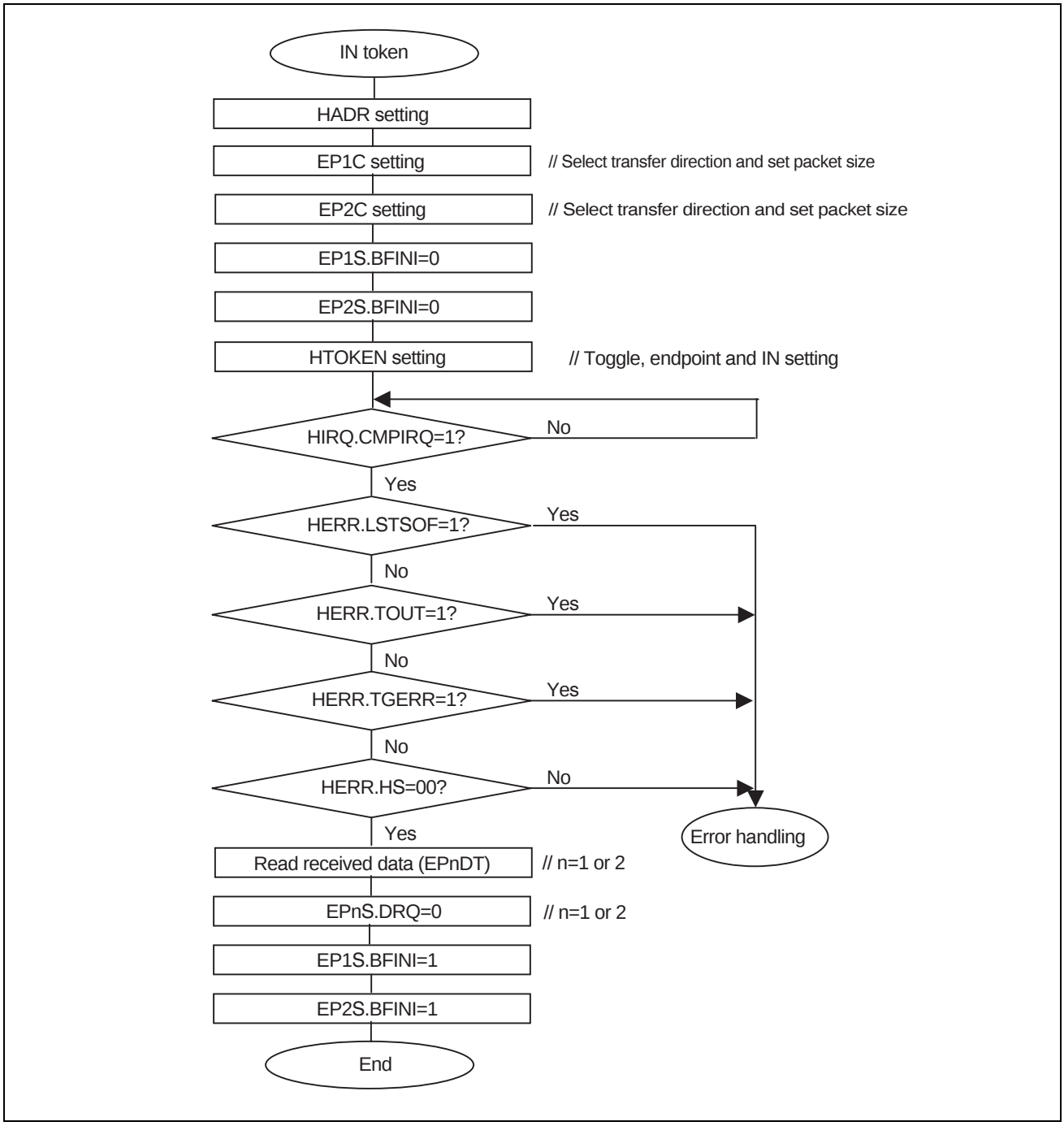
32.5 Flowcharts for tokens in operation with USB Mini-HOST

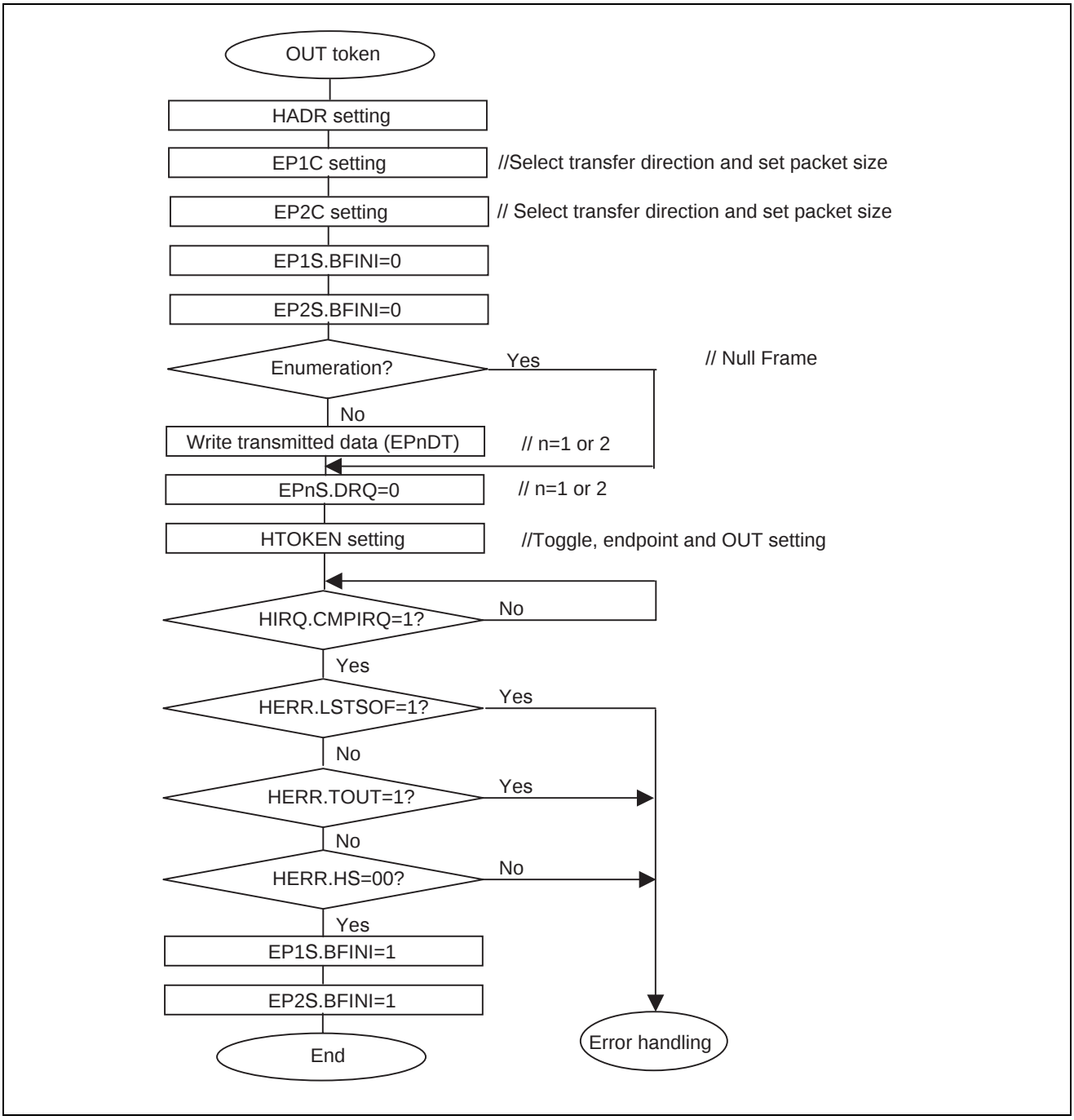
The following figures are flowcharts for tokens in operation with USB Mini-HOST.

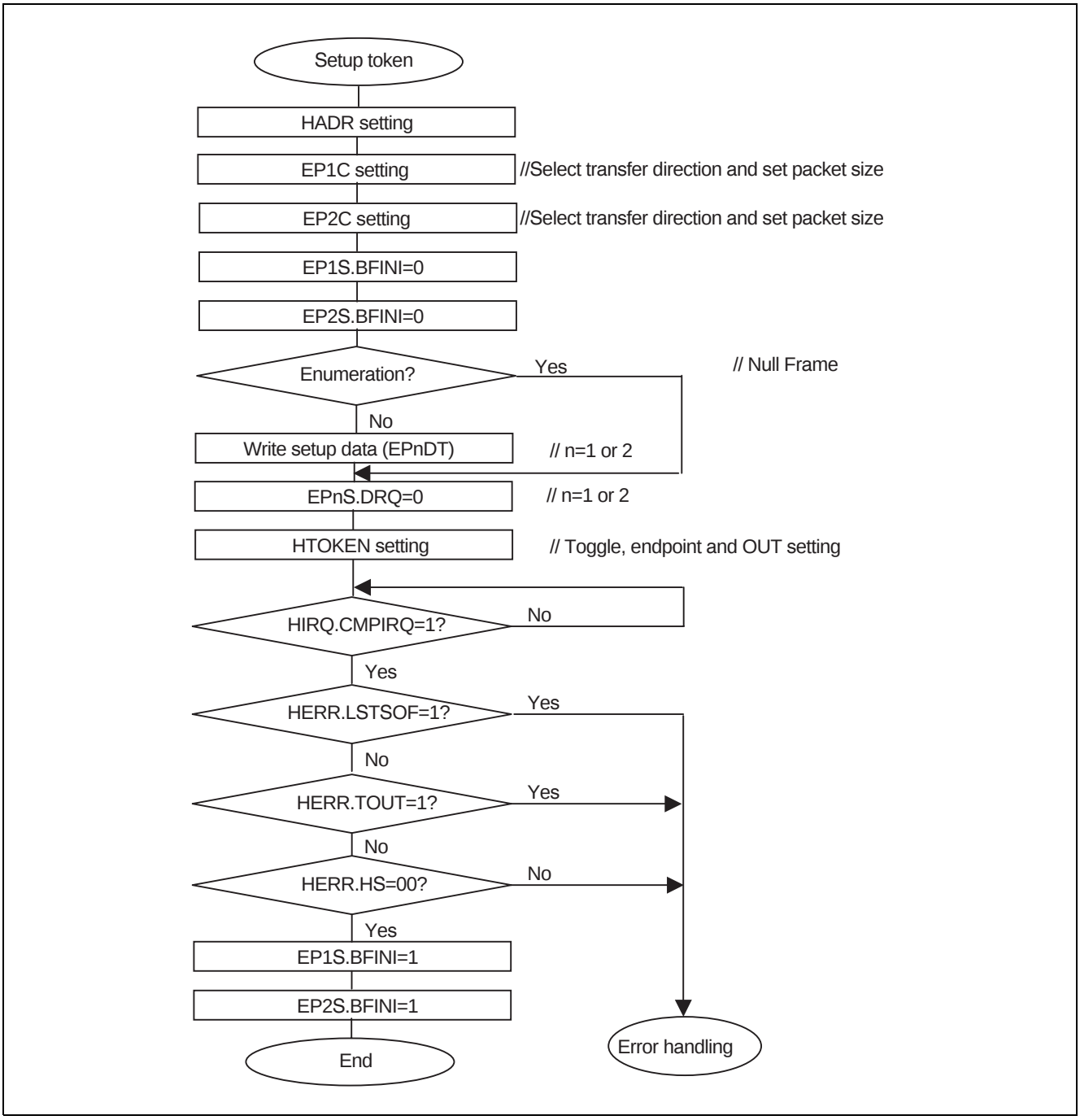
■ Initialization and device detection



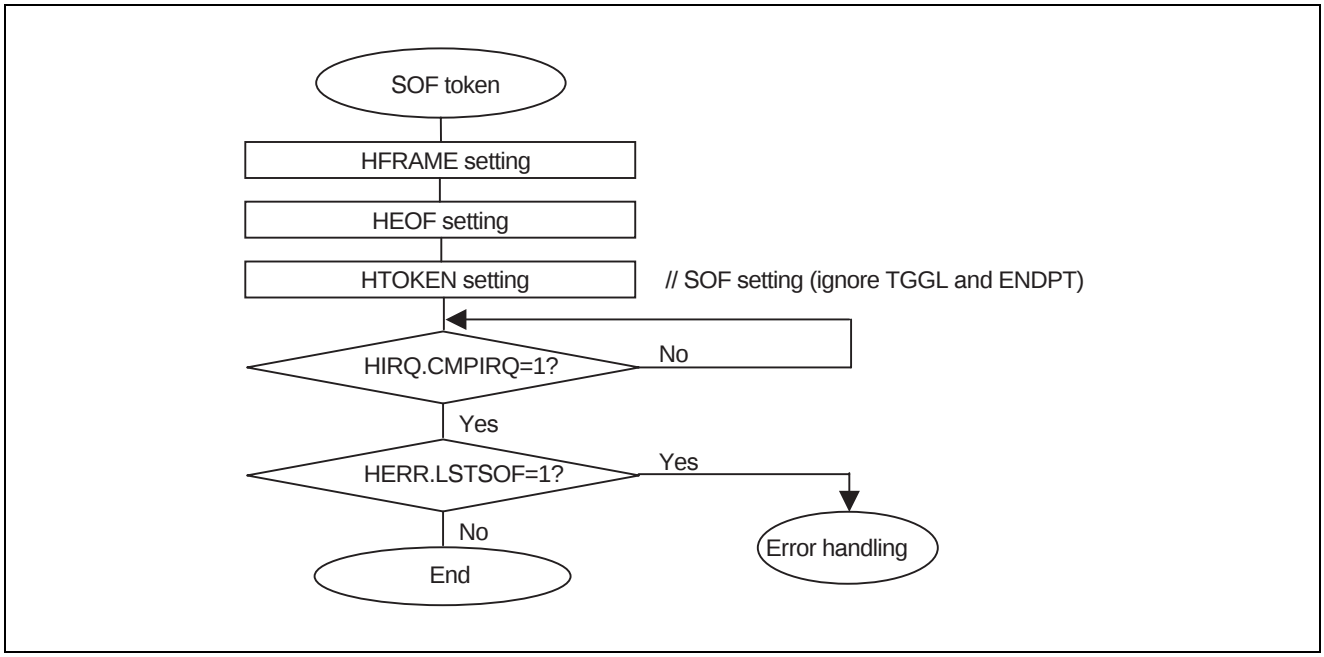
■ IN, OUT, SETUP tokens




OUT token
HADR setting
EP1C setting
//Select transfer direction and set packet size
EP2C setting
// Select transfer direction and set packet size
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write transmitted data (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
//Toggle, endpoint and OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error handling


Setup token
HADR setting
EP1C setting
//Select transfer direction and set packet size
EP2C setting
//Select transfer direction and set packet size
EP1S.BFINI=0
EP2S.BFINI=0
Enumeration?
Yes
// Null Frame
No
Write setup data (EPnDT)
// n=1 or 2
EPnS.DRQ=0
// n=1 or 2
HTOKEN setting
// Toggle, endpoint and OUT setting
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
HERR.TOUT=1?
Yes
No
HERR.HS=00?
No
Yes
EP1S.BFINI=1
EP2S.BFINI=1
End
Error handling

■ SOF token


SOF token
HFRAME setting
HEOF setting
HTOKEN setting
// SOF setting (ignore TGGL and ENDPT)
HIRQ.CMPIRQ=1?
No
Yes
HERR.LSTSOF=1?
Yes
No
End
Error handling

CHAPTER 33 DMA Controller (DMAC)

This chapter explains the functions and operations of the DMA controller (DMAC).

33.1 Overview
33.2 Configuration
33.3 Pins
33.4 Registers
33.5 Interrupts
33.6 An Explanation of Operations and Setting Procedure Examples

33.1 Overview

The DMA controller (DMAC) is used for the DMA (Direct Memory Access) transfer. This controller enhances system performance by enabling high performance of data transfers without going through the CPU.
This series has 8 built-in channels for the DMA controller (DMAC).

■ Overview

This section explains the features of the DMA controller (DMAC).

- Address space: 32-bit (4 GB)
- Transfer mode: One of the following 3 modes can be selected.
  - Block transfer

    This mode is used to transfer 1 block of data when a transfer request is generated. When a transfer request is detected once again after 1 block of data is transferred, the following data is transferred by 1 block. In this mode, the transfer is repeated for the number of times specified.
  - Burst transfer

    In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.
  - Demand transfer

    In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is cancelled or the transfer is completed. If the transfer count is reloaded when the data transfer is completed, the data transfer continues until the transfer request is cancelled.
- Data size: One of the following 3 sizes (widths) can be selected for the data to be transferred.
  - 8 bit
  - 16 bit
  - 32 bit
- Block size: It is possible to select between 1 and 16.
- Transfer count: It is possible to select a figure between 1 and 65535.
- Address update: The transfer source and destination addresses can be updated when data of a specified size (8-bit/16-bit/32-bit) is transferred. One of the following 3 update methods is available.
  - Address increment
  - Address decrement
  - No update (transfer source/destination addresses are fixed)
- Reload function: Whether to reload the following information can be specified upon the completion of data transfer for the specified number of times.
  - Address of transfer source
  - Address of transfer destination
  - Transfer count

- Transfer request: One of the following 4 methods can be selected to generate a transfer request.
  - Generate a transfer request by using software
  - Generate a transfer request by detecting whether there is an interrupt request from a peripheral function.
  - Generate a transfer request by detecting an input from the DREQ0 to DREQ3 pins (only for ch.0 to ch.3)
  - Generate a transfer request by detecting whether there is an interrupt request from on-chip bus IP (slave interface, USB Mini-HOST, USB function)

  The source that generates a transfer request (transfer request source) varies depending on the transfer mode.

  Table 33.1-1 shows the relationship between the transfer mode and transfer request source.

Table 33.1-1 Relationship between the transfer mode and transfer request source.

Transfer Request Source		Block Transfer	Burst Transfer	Demand Transfer
Software		O	O	X
Interrupt request from a peripheral function		O	O	X
DREQ0 to DREQ3 pins		O	O	O
On-chip bus IP	Slave interface	X	X	O
	USB Mini-HOST	O	X	O
	USB function	O	X	O

- Priority: One of the following 2 modes can be selected as having priority when multiple transfer requests are generated.
  - Fixed

    The lowest channel number has priority.

    Order of ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7
  - Round robin

    The following example shows that the channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority.

    Example) Data is transferred from ch.0 to ch.1

    Initial state: ch.0 > ch.1> ch.2> ch.3 > ch.4 > ch.5 > ch.6 > ch.7

    After ch.0 is transferred: ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0

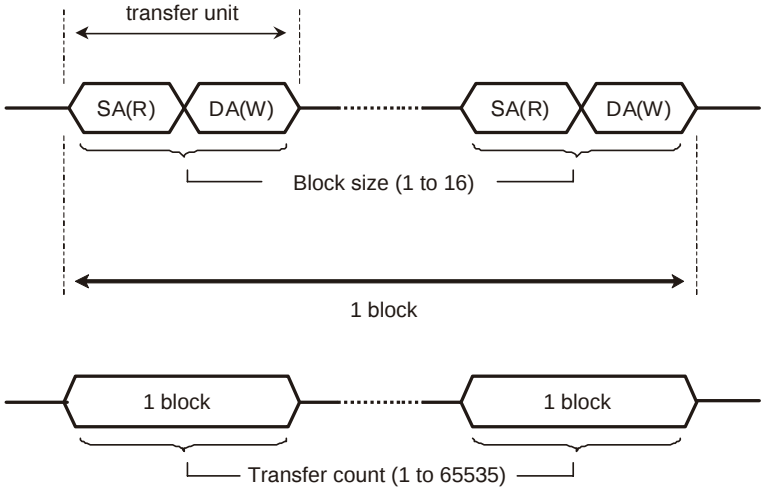
    After ch.1 is transferred: ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1

- Interrupt request: Can be generated in the following cases:
  - Normal end of DMA transfer
  - Abnormal end of DMA transfer
  - Generation of transfer stop requests

■ Definition of Terms

Each term used for the DMA controller (DMAC) is shown in Figure 33.1-1.

Figure 33.1-1 Each Term for the DMA Controller (DMAC)



[transfer unit]

Indicating one minimum transfer of Read (R) from the Source Address (SA) and Write (W) to the Destination Address (DA).

[Block size]

Indicating the transfer count of “1 transfer unit” set with block size bits (bit3 to 0: BLK3 to BLK0) of DMA channel control registers (DCCR0 to DCCR7).

[1 block]

Indicating “1 transfer unit” multiplied by “block size”.

[Transfer count]

Indicating the transfer count of “1 block” set with DMA transfer count registers (DTCR0 to DTCR7).

33.2 Configuration

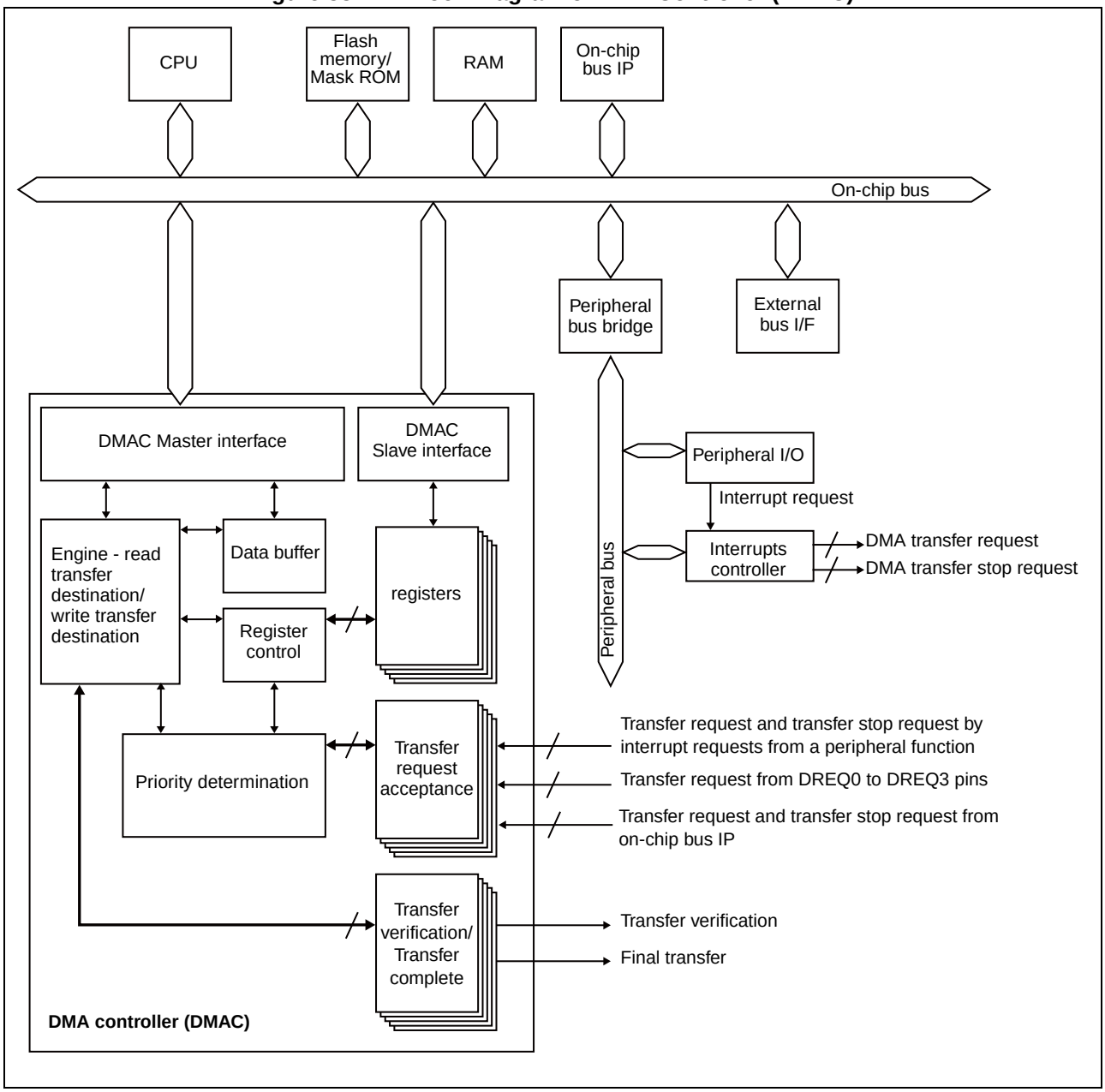
This section explains the DMA controller (DMAC) configuration.

■ Block diagram of DMA controller (DMAC)

Figure 33.2-1 is a block diagram of the DMA controller (DMAC).

The DMA controller (DMAC) is shown in Figure 33.2-1 where it appears inside the frame of DMA controller (DMAC).

Figure 33.2-1 Block Diagram of DMA Controller (DMAC)



- Engine - read transfer destination/write transfer destination

  They are used to read data from the transfer destination of DMA transfer or to write data to the transfer destination.
- Priority determination circuit

  This circuit is used for determining the level of priority of channels for DMA transfers.
- Transfer request acceptance

  Accepts DMA transfer requests.
- Transfer acceptance/transfer complete

  Outputs transfer acceptance or transfer complete.

■ Clocks

Table 33.2-1 shows the clocks used in the DMA controller (DMAC).

Table 33.2-1 Clocks Used in the DMA Controller (DMAC)

Clock Name	Description
Operation clock	On-chip bus clock (HCLK)

33.3 Pins

This section explains the pins used with the DMA controller (DMAC).

■ Overview

The DMA controller (DMAC) has the following pins.

- DREQ0 to DREQ3 pins

  These are the input pins for transfer requests

  For details of these pins, see "CHAPTER 13 External Bus Interface".

  These pins are multiplexed pins. To use these pins as DREQ0 to DREQ3 pins of the DMA controller (DMAC), see "2.4 Setting Method for Pins".
- DACK0 to DACK3 pins

  These are the transfer request acceptance signal output pins.

  For details of these pins, see "CHAPTER 13 External Bus Interface".

  These pins are multiplexed pins. To use these pins as DACK0 to DACK3 pins of the DMA controller (DMAC), see "2.4 Setting Method for Pins".
- DEOP0 to DEOP3 pins

  These are the transfer complete signal output pins.

  For details of these pins, see "CHAPTER 13 External Bus Interface".

  These pins are multiplexed pins. To use these pins as DEOP0 to DEOP3 pins of the DMA controller (DMAC), see "2.4 Setting Method for Pins".

■ Relationship between pins and channels

Table 33.3-1 outlines the relationship between channels and pins.

Table 33.3-1 Relationship between Channels and Pins

Channel	Transfer Request Input Pin	Transfer Request Acceptance Signal Output Pin	Transfer Complete Signal Output Pin
0	DREQ0	DACK0	DEOP0
1	DREQ1	DACK1	DEOP1
2	DREQ2	DACK2	DEOP2
3	DREQ3	DACK3	DEOP3

33.4 Registers

This section explains the configurations and functions of the registers used by the DMA controller (DMAC).

■ List of registers

Table 33.4-1 shows the registers of the DMA controller (DMAC).

Table 33.4-1 Registers of the DMA Controller (DMAC) (1 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	DMACR	DMA Control register	33.4.1
	DILVR	DMA-halt by interrupt level register	33.4.7
0	DCCR0	DMA channel control register 0	33.4.5
	DCSR0	DMA channel status register 0	33.4.6
	DTCR0	DMA transfer count register 0	33.4.4
	DSAR0	DMA source address register 0	33.4.2
	DDAR0	DMA destination address register 0	33.4.3
1	DCCR1	DMA channel control register 1	33.4.5
	DCSR1	DMA channel status register 1	33.4.6
	DTCR1	DMA transfer count register 1	33.4.4
	DSAR1	DMA source address register 1	33.4.2
	DDAR1	DMA destination address register 1	33.4.3
2	DCCR2	DMA channel control register 2	33.4.5
	DCSR2	DMA channel status register 2	33.4.6
	DTCR2	DMA transfer count register 2	33.4.4
	DSAR2	DMA source address register 2	33.4.2
	DDAR2	DMA destination address register 2	33.4.3
3	DCCR3	DMA channel control register 3	33.4.5
	DCSR3	DMA channel status register 3	33.4.6
	DTCR3	DMA transfer count register 3	33.4.4
	DSAR3	DMA source address register 3	33.4.2
	DDAR3	DMA destination address register 3	33.4.3

Table 33.4-1 Registers of the DMA Controller (DMAC) (2 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
4	DCCR4	DMA channel control register 4	33.4.5
	DCSR4	DMA channel status register 4	33.4.6
	DTCR4	DMA transfer count register 4	33.4.4
	DSAR4	DMA source address register 4	33.4.2
	DDAR4	DMA destination address register 4	33.4.3
5	DCCR5	DMA channel control register 5	33.4.5
	DCSR5	DMA channel status register 5	33.4.6
	DTCR5	DMA transfer count register 5	33.4.4
	DSAR5	DMA source address register 5	33.4.2
	DDAR5	DMA destination address register 5	33.4.3
6	DCCR6	DMA channel control register 6	33.4.5
	DCSR6	DMA channel status register 6	33.4.6
	DTCR6	DMA transfer count register 6	33.4.4
	DSAR6	DMA source address register 6	33.4.2
	DDAR6	DMA destination address register 6	33.4.3
7	DCCR7	DMA channel control register 7	33.4.5
	DCSR7	DMA channel status register 7	33.4.6
	DTCR7	DMA transfer count register 7	33.4.4
	DSAR7	DMA source address register 7	33.4.2
	DDAR7	DMA destination address register 7	33.4.3

33.4.1 DMA Control Register (DMACR)

This register controls the entire DMA controller (DMAC).

Figure 33.4-1 shows the bit configuration of the DMA control register (DMACR).

Figure 33.4-1 Bit Configuration of DMA Control Register (DMACR)

bit	31	30	29	28	27	26	25	24
	DME	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
	AT	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Be sure to access this register in units of words.

[bit31]: DME (Operation enable bit of DMA)

This bit enables/disables the entire operation of the DMA controller (DMAC).

Written Value	Explanation
0	Disables the entire operation of the DMA controller (DMAC).
1	Enables the entire operation of the DMA controller (DMAC).

<Notes>

- When "0" is written to this bit to disable the entire operation of the DMA controller (DMAC), DMA transfer cannot be executed even if channel operations are enabled by the CE bit (CE = 1) of DMA channel control registers (DCCR0 to DCCR7).
- If "0" is written to this bit during DMA transfer, the transfer stops when 1 block of the data for transfer has been transferred.

[bit30 to bit16]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit15]: AT (Priority setting bit)

This bit selects one of the following for setting priority if multiple transfer requests are generated.

- Fixed: The lowest channel number is selected.
- Round robin: The priority is determined every time 1 block of data is transferred. The channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority.

  Example) Transferring data from ch.0 to ch.1

  Initial state: ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7

  After ch.0 is transferred: ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0

  After ch.1 is transferred: ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1

Written Value	Explanation
0	Fixed
1	Round robin

<Note>

The priority set by this bit is determined per transfer of the block specified by the BLK3 to BLK0 bits of the DMA channel control registers (DCCR0 to DCCR7).
The priority is not determined during transferring by the demand transfer.

[bit14 to bit0]: Reserved bits

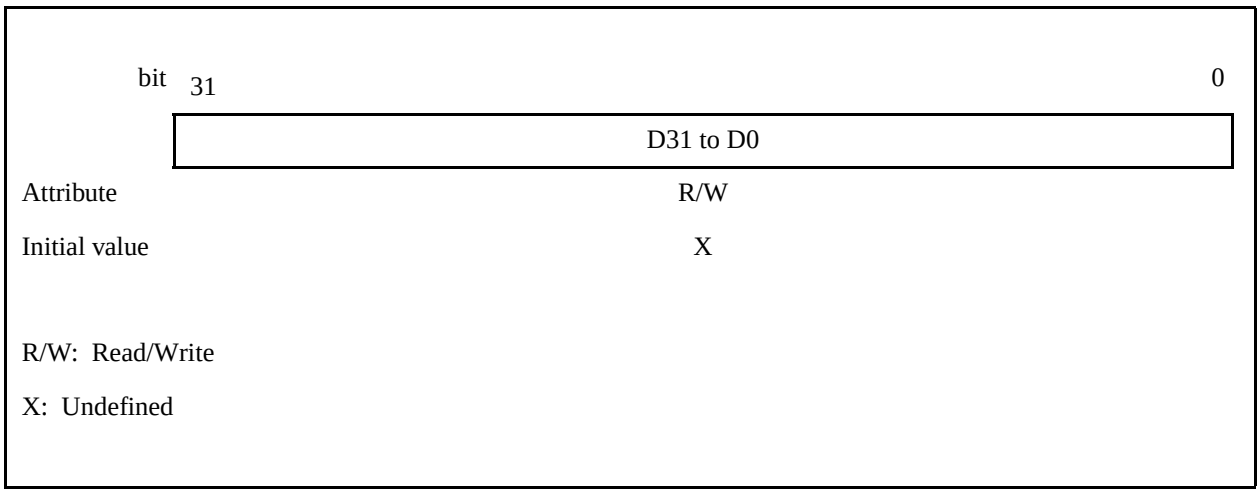
In case of writing	Always write "0" to this bit (these bits).
In case of reading	"0" is read.

33.4.2 DMA Source Address Registers (DSAR0 to DSAR7)

These registers are used to set the transfer source address. These registers are provided for each channel.

Figure 33.4-2 shows the bit configuration of the DMA source address registers (DSAR0 to DSAR7).

Figure 33.4-2 Bit Configuration of the DMA Source Address Registers (DSAR0 to DSAR7)



If the option of updating the transfer source address is selected by the SAC1 and SAC0 bits (SAC1, SAC0 = 00 or 01) of the DMA channel control registers (DCCR0 to DCCR7), the value of this register (address) is updated per completion of a DMA transfer of which size is specified by the TS1 and TS0 bits.

If the transfer of data by the number of blocks specified at DMA transfer count registers (DTCR0 to DTCR7) has completed, the value of this register becomes as follows depending on the SAR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- SAR = 0: The value of this register becomes the next address subsequent to the last-accessed address after transfer completion.
- SAR = 1: After transfer completion, the value of this register returns to the value that is written prior to the transfer.

<Note>

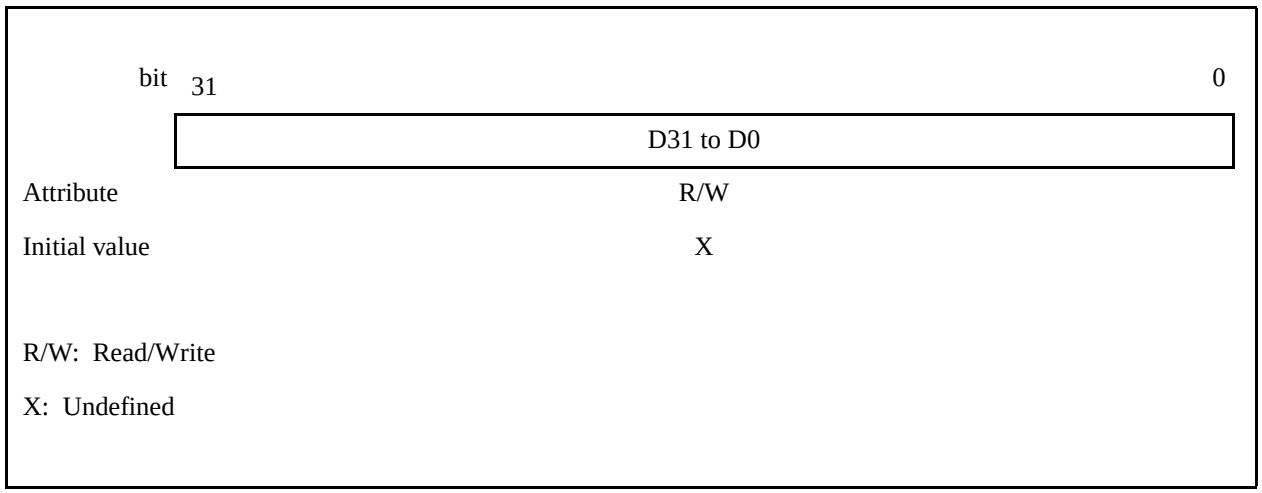
Be sure to access this register in units of words.

33.4.3 DMA Destination Address Registers (DDAR0 to DDAR7)

These registers are used to set the address of transfer destination. These registers are provided for each channel.

Figure 33.4-3 shows the bit configuration of the DMA destination address registers (DDAR0 to DDAR7).

Figure 33.4-3 Bit Configuration of DMA Destination Address Registers (DDAR0 to DDAR7)



If the option of updating transfer destination address is selected by the DAC1 and DAC0 bits (DAC1, DAC0 = 00 or 01) of DMA channel control registers (DCCR0 to DCCR7), the value of this register (address) is updated per completion of a DMA transfer of which size is specified by the TS1 and TS0 bits.

If a transfer of data by the number of blocks specified at DMA transfer count registers (DTCR0 to DTCR7) has completed, the value of this register becomes as follows depending on the DAR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- DAR = 0: The value of this register becomes the next address subsequent to the last-accessed address after transfer completion.
- DAR = 1: After transfer completion, the value of this register returns the value that is written prior to the transfer.

<Note>

Be sure to access this register in units of words.

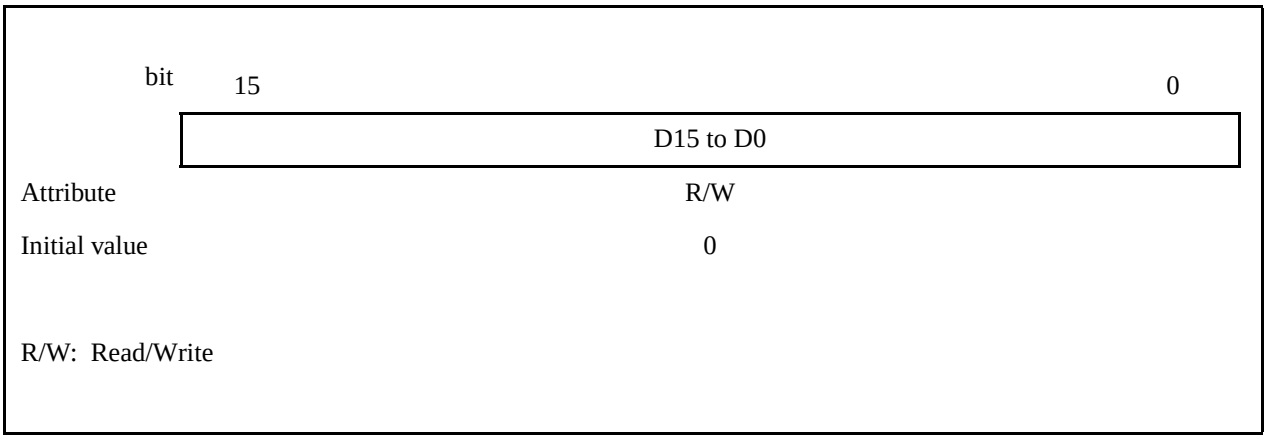
33.4.4 DMA Transfer Count Registers (DTCR0 to DTCR7)

The number of data blocks to transfer can be selected from within the range of 1 to 65535. Also, reading this value shows the number of remaining data blocks to be transferred. These registers are provided for each channel.

The value of these registers decrements by 1 per transfer of 1 block of data. When this register value becomes "0", the transfer ends.

Figure 33.4-4 shows the bit configuration of the DMA transfer count registers (DTCR0 to DTCR7).

Figure 33.4-4 Bit Configuration of DMA Transfer Count Registers (DTCR0 to DTCR7)



If a transfer of data by the number of blocks specified at these registers has completed, the value of this register becomes as follows depending on the TCR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- TCR = 0: The value of this register becomes "0" after transfer completion.
- TCR = 1: After transfer completion, the value of this register returns to the value that is written prior to the transfer.

<Notes>

- Transfer is not executed if "0" is set to this register.
- Be sure to access this register in units of halfwords.
- If the DMA transfer is suspended or ends abnormally, this register shows the number of remaining transfers.

33.4.5 DMA Channel Control Registers (DCCR0 to DCCR7)

These registers control the channels of the DMA controller (DMAC). These registers are provided for each channel.

Figure 33.4-5 shows the bit configuration of the DMA channel control registers (DCCR0 to DCCR7).

Figure 33.4-5 Bit Configuration of the DMA Channel Control Registers (DCCR0 to DCCR7)

bit	31	30	29	28	27	26	25	24
	CE	Reserved	Reserved	Reserved	Reserved	AIE	SIE	NIE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
	Reserved	Reserved	RS1	RS0	Reserved	Reserved	TM1	TM0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
	ST	SAR	SAC1	SAC0	DT	DAR	DAC1	DAC0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	TCR	Reserved	TS1	TS0	BLK3	BLK2	BLK1	BLK0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Be sure to access this register in units of words.

[bit31]: CE (Channel operation enable bit)

This bit enables/disables the operation of channels.

Written Value	Explanation
0	Disables channel operation.
1	Enables channel operation.

If "1" is written to this bit when a request source of DMA transfer is set to software by using the RS1 and RS0 bits (RS1, RS0 = 00), DMA transfer starts. After transfer completion, this bit is automatically cleared to "0".

If RS1 and RS0 bits are set to other than "00", only the channel operation is enabled by writing "1" to this bit.

In this case, if a transfer request specified by the RS1 and RS0 bits is detected, transfer starts. The value of this bit becomes as follows by the TCR bit settings.

- TCR = 0: The bit is cleared to "0" after transfer completion.
- TCR = 1: The bit is not cleared to "0" even after transfer completion.

<Note>

If "0" is written to this bit during the DMA transfer, the transfer stops when 1 block of the data for transfer has been transferred.

In such case, transfer is not re-executed until "1" is written to this bit again and a transfer request is detected.

[bit30 to bit27]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit26]: AIE (Abnormal end interrupt enable bit)

During enabling an abnormal end interrupt of a channel (AIE=1), an abnormal end interrupt request is output when a prohibited value is set to this register.

However, a flag bit (AC bit of DMA channel status registers (DCSR0 to DCSR7)) indicating the abnormal end will be changed to "1" regardless of the settings.

If a value set to this register is any of the following, it is regarded as having an abnormal end of DMA transfer.

- TM1, TM0 bits = 10 (setting prohibited)
- SAC1, SAC0 bits = 10 (setting prohibited)
- DAC1, DAC0 bits = 10 (setting prohibited)
- TS1, TS0 bits = 11 (setting prohibited)
- RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

Written Value	Explanation
0	Disables generation of abnormal end interrupt requests.
1	Enables generation of abnormal end interrupt requests.

<Note>

When AIE=0, writing AIE=1 and setting the prohibited value to the register, it becomes AC=1 and AIE=1 of DMA channel status register (DCSR0 to DCSR7); however, an abnormal end interrupt does not occur.

[bit25]: SIE (Transfer suspension interrupt enable bit)

During enabling a transfer stop interrupt of a channel (SIE=1), an interrupt request is output when a transfer is stopped by the transfer stop request.

However, when the transfer stop request is generated, the flag bit indicating transfer stop by the transfer stop request (SP bit of DMA channel status register (DCSR0 to DCSR7)) changes to "1" regardless of the settings.

Written Value	Explanation
0	Disables generation of transfer suspension interrupt requests.
1	Enables generation of transfer suspension interrupt requests.

[bit24]: NIE (Normal end interrupt enable bit)

During enabling a normal end interrupt of a channel (NIE=1), an interrupt request is output when DMA transfer normally ends.

However, the flag bit indicating normal end (NC bit (DCSR0 to DCSR7) of DMA channel status register) changes to "1" regardless of the settings when DMA transfer normally ends.

Under any of the following conditions, DMA transfer is regarded as having a normal end.

- Transfer completes for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).
- If "0" is set for the value of DMA transfer count registers (DTCR0 to DTCR7), channel operations are enabled by the CE bit (CE = 1).

Written Value	Explanation
0	Disables generation of normal end interrupt requests.
1	Enables generation of normal end interrupt requests.

[bit23, bit22]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit21, bit20]: RS1, RS0 (Transfer request source bits)

These bits set the source that generates a transfer request (transfer request source) from any of the following 4 options.

However, do not set "10" for ch.4 to ch.7 because the DREQ0 to DREQ3 pins can be selected only with ch.0 to ch.3

- Generate a transfer request by using software.
- Generate a transfer request by detecting an interrupt request from a peripheral function.
- Generate a transfer request by detecting an input from the DREQ0 to DREQ3 pins.
- Generate a transfer request by detecting whether there is an interrupt request from on-chip bus IP (slave interface, USB Mini-HOST, and USB function)

RS1	RS0	Explanation
0	0	Software
0	1	Interrupt request from a peripheral function
1	0	DREQ0 to DREQ3 pins (ch.0 to ch.3 only)
1	1	On-chip bus IP

<Notes>

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), only the DREQ0 to DREQ3 pins can be set for the transfer request source.
- If interruption requests from a peripheral function are set as the transfer request source, the following register settings are required.
  - IO-data request registers (IORR0 to IORR7)
  - Select registers for DMA transfer request clear by a peripheral (ICSEL0 to ICSEL14)

  See "CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function" for details of each register.
- If the DREQ0 to DREQ3 pins are set as the transfer request source, the DMA transfer setting of the external bus interface is required. See "CHAPTER 13 External Bus Interface".
- If the on-chip bus IP is set as the transfer request source, the DMA transfer setting for it is required. See "CHAPTER 14 Slave Interface" for the slave interface, "CHAPTER 32 USB Mini-HOST" for USB Mini-HOST, and "CHAPTER 31 USB Function" for the USB function.

[bit19, bit18]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit17, bit16]: TM1, TM0 (Transfer mode bits)

These bits set the transfer mode from any of the following 3 modes.

- Block transfer

  This mode is used to transfer 1 block of data when a transfer request is generated. When a transfer request is detected again after 1 block of data is transferred, the following data is transferred by 1 block. In this mode, the transfer is repeated for the number of times specified.

- Burst transfer

  In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.

- Demand transfer

  In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is cancelled or the transfer is completed. If the reload option of transfer count is enabled at the completion of the data transfer, the data transfer continues until the transfer request is cancelled.

TM1	TM0	Explanation
0	0	Block transfer
0	1	Burst transfer
1	0	Setting prohibited
1	1	Demand transfer

<Note>

When demand transfer is selected, both or either of the transfer source/transfer destination needs to be set as other than memory (ST/DT = 1) by using the ST bit or DT bit.

[bit15]: ST (Transfer source type bits)

This bit sets whether the transfer source data exists in the memory.

Written Value	Explanation
0	Memory
1	Other than memory (such as a peripheral function)

If this bit is set to"1", it outputs a transfer complete signal that indicates the end of a transfer or a transfer request acceptance signal for the transfer request in the reading transfer source cycle.

<Notes>

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), both or either of transfer source/transfer destination need to be set at other than memory.
  When demand transfer is selected, set "1" to both or either of this bit/DT bit.
- In case of performing the DMA transfer with the USB function or USB Mini-HOST, always set the ST bit to "0" regardless of the transfer source type.

[bit14]: SAR (Transfer source address reload bit)

This bit specifies whether to return the value of DMA source address registers (DSAR0 to DSAR7) to the value before transfer upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer source address).

Written Value	Explanation
0	Disables reload. The value of DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	Enables reload. After transfer completion, the value of DMA source address registers (DSAR0 to DSAR7) returns to the value that is written prior to the transfer.

[bit13, bit12]: SAC1, SAC0 (Transfer source address count bits)

These bits specify whether to update the value of DMA source address registers (DSAR0 to DSAR7) from one of the following 3 options per completion of data transfer of the size specified with TS1 and TS0 bits.

SAC1	SAC0	Explanation
0	0	Address increment
0	1	Address decrement
1	0	Setting prohibited
1	1	Address fixed

If the setting of address increment/address decrement is selected, the increment/decrement value varies depending on the transfer size specified with TS1 and TS0 bits.

Table 33.4-2 shows the relationship between the transfer size and the address increment/decrement value.

Table 33.4-2 Relationship between the transfer size and the value of address increment / decrement

Transfer Size	Increment/Decrement Value
8 bit	1
16 bit	2
32 bit	4

[bit11]: DT (Transfer destination type bits)

This bit sets the transfer destination.

Written Value	Explanation
0	Memory
1	Other than memory (such as a peripheral function)

If this bit is set to "1", it can output a transfer request acceptance signal or a transfer complete signal that indicates an end of transfer in the cycle of writing to the transfer destination.

<Notes>

- If transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11), both or either of transfer source/transfer destination need to be set at other than memory.
  When demand transfer is selected, set "1" to both or either of this bit/ST bit.
- In case of performing the DMA transfer with the USB function or USB Mini-HOST, always set the DT bit to "1" regardless of the transfer destination type.

[bit10]: DAR (Transfer destination address reload bits)

This bit specifies whether to return the value of DMA destination address registers (DDAR0 to DDAR7) to the value before transfer upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer source address).

Written Value	Explanation
0	Disables reload. The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	Enables reload. After transfer completion, the value of DMA destination address registers (DDAR0 to DDAR7) returns to the value that is written prior to the transfer.

[bit9, bit8]: DAC1, DAC0 (Transfer destination address count bits)

These bits specify whether to update the value of DMA destination address registers (DDAR0 to DDAR7) from one of the following 3 options per completion of data transfer of the size specified with TS1 and TS0 bits.

DAC1	DAC0	Explanation
0	0	Address increment
0	1	Address decrement
1	0	Setting prohibited
1	1	Address fixed

If the setting of address increment/address decrement is selected, the increment/decrement value varies depending on the transfer size specified with TS1 and TS0 bits.

Table 33.4-3 shows the relationship between the transfer size and the address increment/decrement value.

Table 33.4-3 Relationship between the Transfer Size and the Value of Address Increment/ Decrement

Transfer Size	Increment/Decrement Value
8 bit	1
16 bit	2
32 bit	4

[bit7]: TCR (Transfer count reload bit)

This bit specifies whether to have the specified transfer count reloaded to DMA transfer count registers (DTCR0 to DTCR7) upon the completion of data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) (enable/disable reload of transfer count).

Written Value	Explanation
0	Disables reload. After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) is cleared to "0".
1	Enables reload. After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) returns to the value that is written prior to the transfer.

<Notes>

- If "1" is set to this bit and the transfer request source is set to other than software by RS1 and RS0 bits, the CE bit is not cleared to "0" and enters the transfer request wait state even after transfer completion.
- If "0" is written to this bit, the CE bit is automatically cleared to "0" after transfer completion regardless of the transfer request source.
- If the reload is enabled by writing "1" to this bit, transfer continues sequentially in demand transfer mode while transfer requests are being output irrespective of whether transfer has completed for the number of times specified.

[bit6]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit5, bit4]: TS1, TS0 (Transfer size bits)

These bits set the size (width) of data to be transferred at one time from any of the following 3 modes.

TS1	TS0	Explanation
0	0	8 bit
0	1	16 bit
1	0	32 bit
1	1	Setting prohibited

[bit3 to bit0]: BLK3 to BLK0 (Block size bits)

These bits are used to specify the number of times for 1 transfer unit (size) within 1 block.

BLK3	BLK2	BLK1	BLK0	Explanation
0	0	0	0	1 time
0	0	0	1	2 times
0	0	1	0	3 times
0	0	1	1	4 times
0	1	0	0	5 times
0	1	0	1	6 times
0	1	1	0	7 times
0	1	1	1	8 times
1	0	0	0	9 times
1	0	0	1	10 times
1	0	1	0	11 times
1	0	1	1	12 times
1	1	0	0	13 times
1	1	0	1	14 times
1	1	1	0	15 times
1	1	1	1	16 times

33.4.6 DMA Channel Status Registers (DCSR0 to DCSR7)

These registers show the state of DMA controller (DMAC). These registers are provided for each channel.

Figure 33.4-6 shows the bit configuration of the DMA channel status registers (DCSR0 to DCSR7).

Figure 33.4-6 Bit Configuration of DMA Channel Status Registers (DCSR0 to DCSR7)

bit	15	14	13	12	11	10	9	8
	CA	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	AC	SP	NC
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

R: Read only

<Note>

Be sure to access this register in units of half words.

[bit15]: CA (Channel active bit)

This bit indicates the operating state of channels.

Read Value	Explanation
0	The channel is inactive.
1	The channel is active.

<Notes>

- When "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7), this bit is changed to "1".
- This bit is changed to "0" in any of the following cases:
  - Transfer ends.
  - "0" is written to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).

[bit14 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit2]: AC (Abnormal end state flag bit)

This bit indicates an abnormal end of DMA transfer.

If the value of DMA channel control registers (DCCR0 to DCCR7) is any of the following, DMA transfer is regarded as having an abnormal end.

- TM1, TM0 bits = 10 (setting prohibited)
- SAC1, SAC0 bits = 10 (setting prohibited)
- DAC1, DAC0 bits = 10 (setting prohibited)
- TS1, TS0 bits = 11 (setting prohibited)
- RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

If "1" is set to the AIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", an abnormal end interrupt request is generated.

AC	In Case of Reading	In Case of Writing
0	No abnormal end is detected.	This bit is cleared to "0".
1	An abnormal end is detected.	Ignored

<Note>

This bit is not cleared automatically. To clear the abnormal end interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.

[bit1]: SP (Transfer suspension state flag bit)

This bit indicates a suspension of transfer due to a transfer stop request from the transfer request source.

If "1" is set to the SIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", a transfer suspension interrupt request is generated.

SP	In Case of Reading	In Case of Writing
0	Transfer is not suspended.	This bit is cleared to "0".
1	Transfer is suspended.	Ignored

<Note>

This bit is not cleared automatically. To clear the transfer suspension interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.

[bit0]: NC (Normal end state flag bit)

This bit indicates a normal end of DMA transfer.

Under any of the following conditions, the DMA transfer is regarded as ending normally.

- Transfer completes for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).
- If "0" is set for the value of DMA transfer count registers (DTCR0 to DTCR7), channel operations are enabled by the CE bit (CE = 1) of DMA channel control registers (DCCR0 to DCCR7).

If "1" is set to the NIE bit of the DMA channel control registers (DCCR0 to DCCR7) when this bit is set to "1", a normal end interrupt request is generated.

NC	In Case of Reading	In Case of Writing
0	No normal end of transfer is detected.	This bit is cleared to "0".
1	A normal end of transfer is detected.	Ignored

<Note>

This bit is not cleared automatically. To clear the normal end interrupt request flag, write "0" to this bit before enabling the DMA transfer operation.

33.4.7 DMA-Halt by Interrupt Level Register (DILVR)

This register specifies whether to halt DMA transfer if an interrupt request is generated from a peripheral function.

Figure 33.4-7 shows the bit configuration of DMA-halt by interrupt level register (DILVR).

Figure 33.4-7 Bit Configuration of the DMA-Halt by Interrupt Level Register (DILVR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	LVL4	LVL3	LVL2	LVL1	LVL0
Attribute	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial value	0	0	0	1	1	1	1	1

R/W: Read/Write

R: Read only

<Note>

Be sure to access this register in units of bytes.

[bit7 to bit5]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit4 to bit0]: LVL4 to LVL0 (DMA-halt by interrupt level bits)

These bits specify the interrupt level where DMA transfer is halted. If an interrupt request of a level that is higher than the interrupt level specified with this bit is generated from the peripheral functions, DMA transfer is halted.

LVL4	LVL3	LVL2	LVL1	LVL0	Interrupt Request Level Where DMA Transfers Are Halted
1	0	0	0	0	DMA transfers are not halted.
1	0	0	0	1	Interrupt request of "11_H" or higher
1	0	0	1	0	Interrupt request of "12_H" or higher
1	0	0	1	1	Interrupt request of "13_H" or higher
1	0	1	0	0	Interrupt request of "14_H" or higher
1	0	1	0	1	Interrupt request of "15_H" or higher
1	0	1	1	0	Interrupt request of "16_H" or higher
1	0	1	1	1	Interrupt request of "17_H" or higher
1	1	0	0	0	Interrupt request of "18_H" or higher
1	1	0	0	1	Interrupt request of "19_H" or higher
1	1	0	1	0	Interrupt request of "$1A_H$" or higher
1	1	0	1	1	Interrupt request of "$1B_H$" or higher
1	1	1	0	0	Interrupt request of "$1C_H$" or higher
1	1	1	0	1	Interrupt request of "$1D_H$" or higher
1	1	1	1	0	Interrupt request of "$1E_H$" or higher
1	1	1	1	1	All interrupt requests

<Note>

LVL4 bit is fixed to "1", and only LVL3 to LVL0 can be set.

33.5 Interrupts

An interrupt request is generated under the following conditions:

- Normal end of DMA transfer (normal end interrupt request)
- Abnormal end of DMA transfer (abnormal end interrupt request)
- Suspension of DMA transfer due to the generation of a transfer stop request (transfer suspension interrupt request)

Table 33.5-1 outlines the interrupts that can be used with the DMA controller (DMAC).

Table 33.5-1 Interrupts of the DMA Controller (DMAC)

Interrupt Request	Interrupt Request Flag	Interrupt Request Enabled	Interrupt Clear
Normal end interrupt request	NC of DCSR = 1	NIE of DCCR = 1	Write "0" to the NC bit of the DCSR.
Abnormal end interrupt request	AC of DCSR = 1	AIE of DCCR = 1	Write "0" to the AC bit of the DCSR.
Transfer suspension interrupt request	SP of DCSR = 1	SIE of DCCR = 1	Write "0" to the SP bit of the DCSR.

DCSR: DMA channel status registers (DCSR0 to DCSR7)

DCCR: DMA channel control registers (DCCR0 to DCCR7)

<Notes>

- Clear the interrupt requests after disabling generation of interrupt requests, or clear the interrupt requests in the interrupt processing routine.
- For details of the interrupt vector number of each interrupt request, see "APPENDIX C Interrupt Vectors".
- Use the interrupt control registers (ICR00 to ICR47) to set the interrupt level corresponding to the interrupt vector number. For information on the interrupt level settings, see "CHAPTER 10 Interrupt Controller".

33.6 An Explanation of Operations and Setting Procedure Examples

This section explains the operations of the DMA controller (DMAC). This section also includes examples of procedures for setting the transfer mode.

33.6.1 Transfer Settings

This section explains the settings required for the use of the DMA controller (DMAC).

■ Overview

Settings of the entire DMA controller (DMAC) and the channels to be used for the transfer are required to use the DMA transfer.

To select an interrupt request of a peripheral function as a transfer request source of a DMA transfer, the settings of the interrupt vector number and each peripheral function are also required. For details, see "CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function".

The setup procedure is as follows:

1. Settings of the entire DMA controller (DMAC)
   - See "■ Settings of the entire DMA Controller (DMAC)".
2. Settings of channels to be used for the DMA controller
   - See "■ Channel Settings".

■ Settings of the entire DMA Controller (DMAC)

Settings of the entire DMA controller (DMAC) and the channels to be used for the transfer are required for the use of the DMA controller (DMAC).

This paragraph explains the required settings of the entire DMA controller (DMAC).

- Enabling operation of the DMA controller (DMAC) by the DME bit of DMA control register (DMACR)

  Operation disabled: DME = 0

  Operation enabled: DME = 1
- Settings of priority in AT bit of DMA control register (DMACR)

  Fixed: AT = 0

  Round robin: AT = 1
- Settings of interrupt levels of DMA transfer halts by the LVL4 to LVL0 bits of DMA-halt by interrupt level register (DILVR)

For details, see the explanation for each register.

<Note>

To select the interrupt request generated from a peripheral function as the source that generates DMA transfer requests, select an interrupt vector number before configuring DMA controller (DMAC) settings.

For details of selecting interrupt vector numbers, see "34.3.1 IO-Data Request Registers (IORR0 to IORR7)" in "CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function".

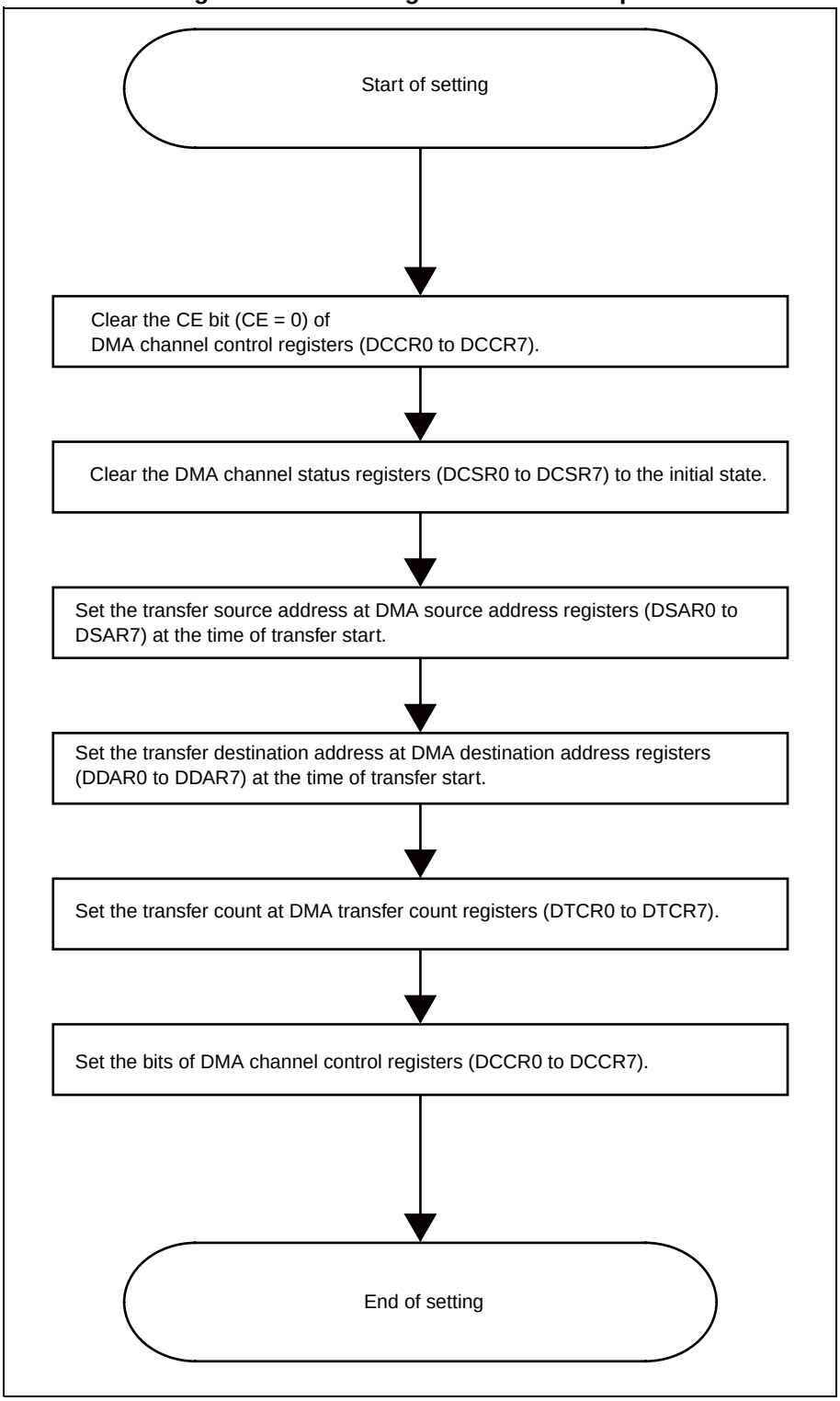
■ Channel Settings

This paragraph explains the settings required for the channels to be used.

Configure channel settings upon the completion of the entire DMA controller (DMAC) settings.

Figure 33.6-1 shows the example of the procedure for configuring channel settings.

Figure 33.6-1 Setting Procedure Examples


Start of setting
Clear the CE bit (CE = 0) of
DMA channel control registers (DCCR0 to DCCR7).
Clear the DMA channel status registers (DCSR0 to DCSR7) to the initial state.
Set the transfer source address at DMA source address registers (DSAR0 to DSAR7) at the time of transfer start.
Set the transfer destination address at DMA destination address registers (DDAR0 to DDAR7) at the time of transfer start.
Set the transfer count at DMA transfer count registers (DTCR0 to DTCR7).
Set the bits of DMA channel control registers (DCCR0 to DCCR7).
End of setting

1. Disable channel operation.

   The CE bit of DMA channel control registers (DCCR0 to DCCR7) is 0.
2. Initialize the flag that indicates the channel state.

   The AC bit / SP bit / NC bit of DMA channel status registers (DCSR0 to DCSR7) is 0.
3. Set the transfer source address.

   D31 to D0 bits of DMA source address registers (DSAR0 to DSAR7)
4. Set the transfer destination address.

   D31 to D0 bits of DMA destination address registers (DDAR0 to DDAR7)
5. Set the transfer count within the range of 1 to 65535 times.

   D15 to D0 bits of DMA transfer count registers (DTCR0 to DTCR7) is 1 or more.
6. Other settings

   Set the bits of DMA channel control registers (DCCR0 to DCCR7).
7. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).

   Channel operation is enabled.

   The transfer is started at the same time that channel operations are enabled if the transfer request source is set to software.

<Note>

For information on procedures for register settings, see the explanation for each register.

33.6.2 Transfer Operations

This section explains the transfer operations of the DMA controller (DMAC).

■ Transfer Mode

The DMA controller (DMAC) has the following 3 transfer modes.

- Block transfer

  This mode is used to transfer 1 block of data when a transfer request is generated. This mode repeats the transfer of the following data by 1 block by the specified number of times when a transfer request is detected once again after 1 block of data is transferred.

  The data of which size is specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are transferred once at a time for the number of times specified by the BLK3 to BLK0 bits.
- Burst transfer

  In this mode, once a transfer request is generated, data is sequentially transferred 1 block at a time until the entire data transfer is completed.

  The data of which size is specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are transferred for the count specified at the BLK3 to BLK0 bits at a time sequentially for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7).
- Demand transfer

  In this mode, once a transfer request is generated, data is transferred sequentially either until the transfer request is cancelled or the transfer is completed. If the reload option of the transfer count is on when the data transfer is completed, the data transfer continues until the transfer request is cancelled.

  In the demand transfer, either or both of transfer source/transfer destination needs to be set to other than memory.

The source that generates a transfer request varies depending on the transfer mode.

Table 33.6-1 shows the relationship between the transfer mode and transfer request source.

Table 33.6-1 Relationship between the transfer mode and transfer request source

Transfer Request Source		Block Transfer	Burst Transfer	Demand Transfer
Software		O	O	X
Interrupt request from a peripheral function		O	O	X
DREQ0 to DREQ3 pins*1		O	O	O
On-chip bus IP	Slave interface	X	X	O
	USB Mini-HOST	O	X	O
	USB function	O	X	O

*1 can be used with ch.0 to ch.3 only.

■ Detection of transfer requests

The transfer operation starts by detecting DMA transfer requests.

Detections of transfer requests vary depending on the transfer request source specified by the RS1 and RS0 bits of DMA channel control registers (DCCR0 to DCCR7).

- The transfer request source is software:

  If "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7), it determines the priority levels of channels and starts the transfer.
- Transfer source is other than software:

  When the CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "1", channel operations are enabled.

  If a transfer request is detected during that state, it determines the priority levels of channels and starts the transfer.

<Notes>

- If interrupt requests of a peripheral function are set as the transfer request source, it is necessary to select an interrupt vector. See "34.3.1 IO-Data Request Registers (IORR0 to IORR7))" in CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function.
- If interrupt requests of a peripheral function are set as transfer request source, set the value of interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) as follows when interrupt requests are generated from a peripheral function.

  ILM less than or equal to ICR
- If the DREQ0 to DREQ3 pins are set as the transfer request source, the DMA transfer setting of the external bus interface is required. See "CHAPTER 13 External Bus Interface".
- If the on-chip bus IP is set as the transfer request source, the DMA transfer setting for it is required. See "CHAPTER 14 Slave Interface" for the slave interface, "CHAPTER 32 USB Mini-HOST" for USB Mini-HOST, and "CHAPTER 31 USB Function" for the USB function.

Table 33.6-2 shows the detect condition of transfer requests and transfer request source.

Table 33.6-2 Detect condition of transfer requests and transfer request source

<table>
<tr><th colspan="2">Transfer Request Source</th><th>Block Transfer</th><th>Burst Transfer</th><th>Demand Transfer</th></tr>
<tr><td colspan="2">Software</td><td colspan="2">Write "1" to the CE bit of the DCCR.</td><td>-</td></tr>
<tr><td colspan="2">Interrupt request</td><td colspan="2" rowspan="2">Edge detection</td><td>-</td></tr>
<tr><td colspan="2">DREQ0 to DREQ3 pins</td><td rowspan="4">First time:
Edge detection
Second time or after:
Level detection</td></tr>
<tr><td rowspan="3">On-chip bus IP</td><td>Slave interface</td><td colspan="2">-</td></tr>
<tr><td>USB Mini-HOST</td><td rowspan="2">Edge detection</td><td rowspan="2">-</td></tr>
<tr><td>USB function</td></tr>
</table>

DCCR: DMA channel control registers (DCCR0 to DCCR7)

■ Operation

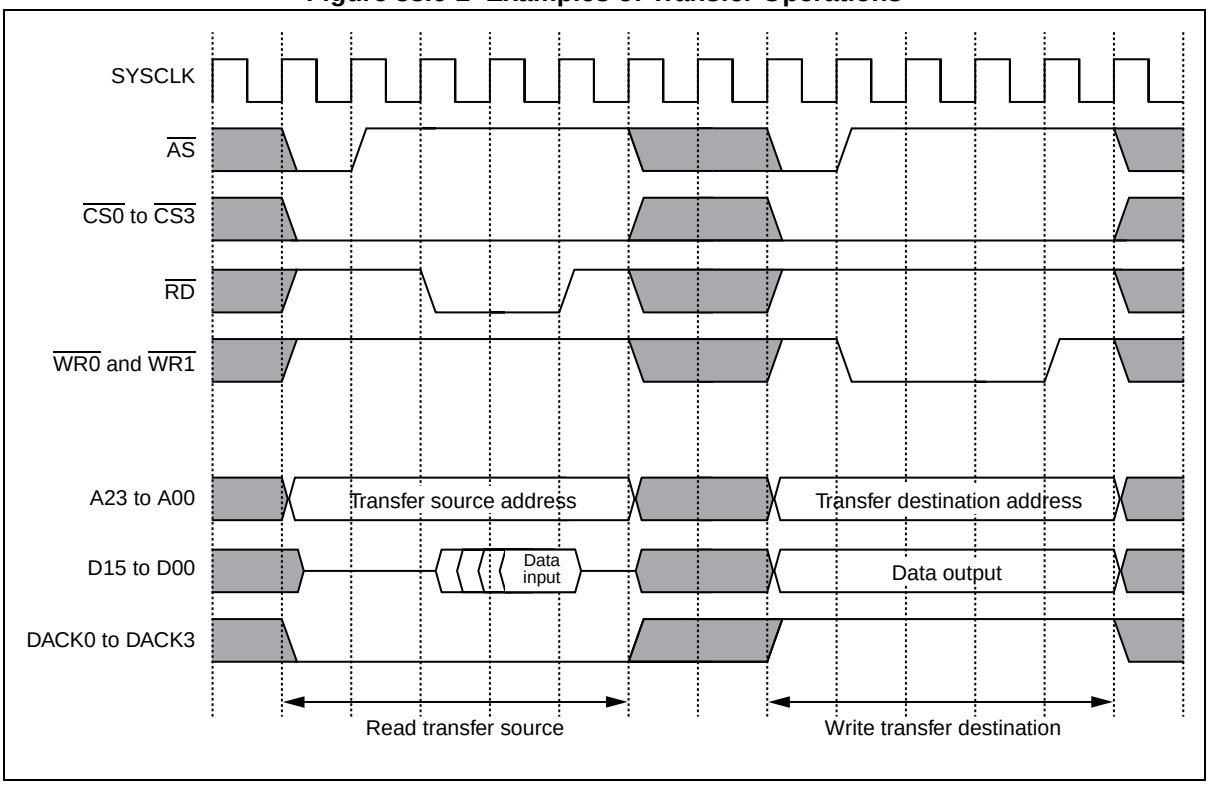
If a transfer request is detected, transfer is executed as follows:

1. Read data from the address specified at DMA source address registers (DSAR0 to DSAR7).

   Data of the bit width specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7) are read.
2. Write data to the address specified by DMA destination address registers (DDAR0 to DDAR7).

Figure 33.6-2 shows examples of the transfer operation between the external memory and external I/O.

Figure 33.6-2 Examples of Transfer Operations



<Note>

For details of these pins in Figure 33.6-2, see "CHAPTER 13 External Bus Interface".

- AS pin
- CS0 to CS3 pins
- RD pin
- WR0 and WR1 pins
- A23 to A00 pins
- D15 to D00 pins
- DACK0 to DACK3 pins

■ Priority

If multiple DMA transfer requests are generated, transfer is executed from the channel with the highest priority specified by the AT bit settings of DMA control registers (DMACR).

Priority is determined every time 1 block of data is transferred. Also, priority is determined at the end of transfer.

<Note>

Priority is not determined during the transfer in demand transfer mode.

Procedure of priority determination is any of the following:

- Fixed: The lowest channel number is selected.

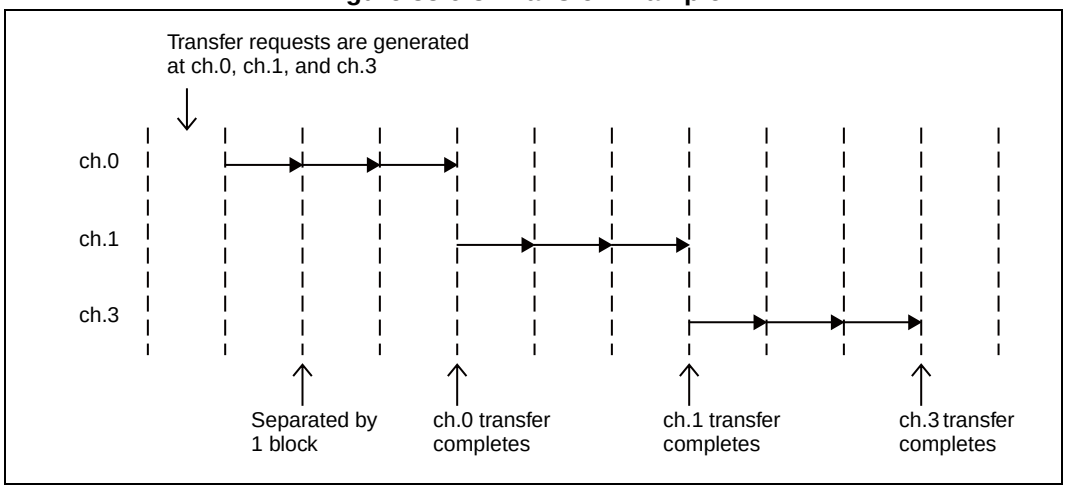
  Figure 33.6-3 shows an example of transfer as transfer example 1 under the following conditions.

  Transfer request: Occurs simultaneously at ch.0, ch.1, and ch.3

  Transfer mode: Burst transfer mode in all channels

  Transfer count: 3 for all channels

Figure 33.6-3 Transfer Example 1



1. Transfer requests are generated simultaneously at ch.0, ch.1, and ch.3.
2. Transfer at ch.0 starts.
3. Once data at ch.0 is transferred for 3 blocks, transfer at ch.1 starts.
4. Once data at ch.1 is transferred for 3 blocks, transfer at ch.3 starts.

Figure 33.6-4 shows an example of transfer as transfer example 2 under the following conditions.

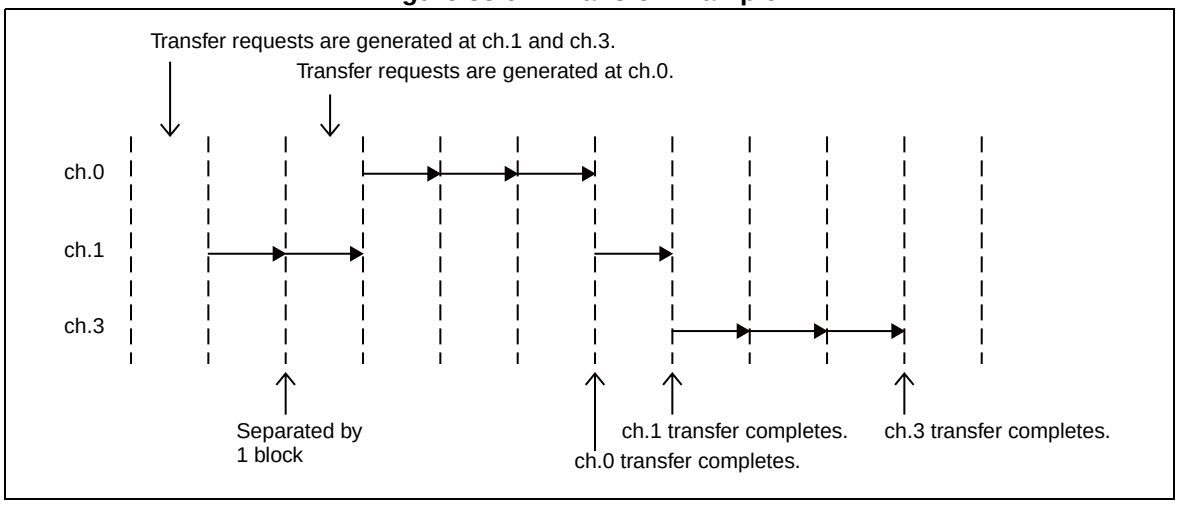
Transfer request:

- Occurs simultaneously at ch.1 and ch.3.
- Occurs at ch.0 during the transfer at ch.1.

Transfer mode: All channels are in burst transfer mode.

Transfer count: 3 for all channels.

Figure 33.6-4 Transfer Example 2



1. Transfer requests are generated simultaneously at ch.1 and ch.3.
2. Transfer at ch.1 starts.
3. Transfer requests are generated at ch.0 during the transfer at ch.1.
4. Transfer at ch.1 is suspended and transfer at ch.0 starts.
5. Once data at ch.0 is transferred for 3 blocks, transfer at ch.1 starts.
6. Once data at ch.1 is transferred for 3 blocks, transfer at ch.3 starts.

- Round robin: The channel which started a transfer became the one with the lowest priority and the channels placed lower than it are moved up in terms of priority one priority level at a time.

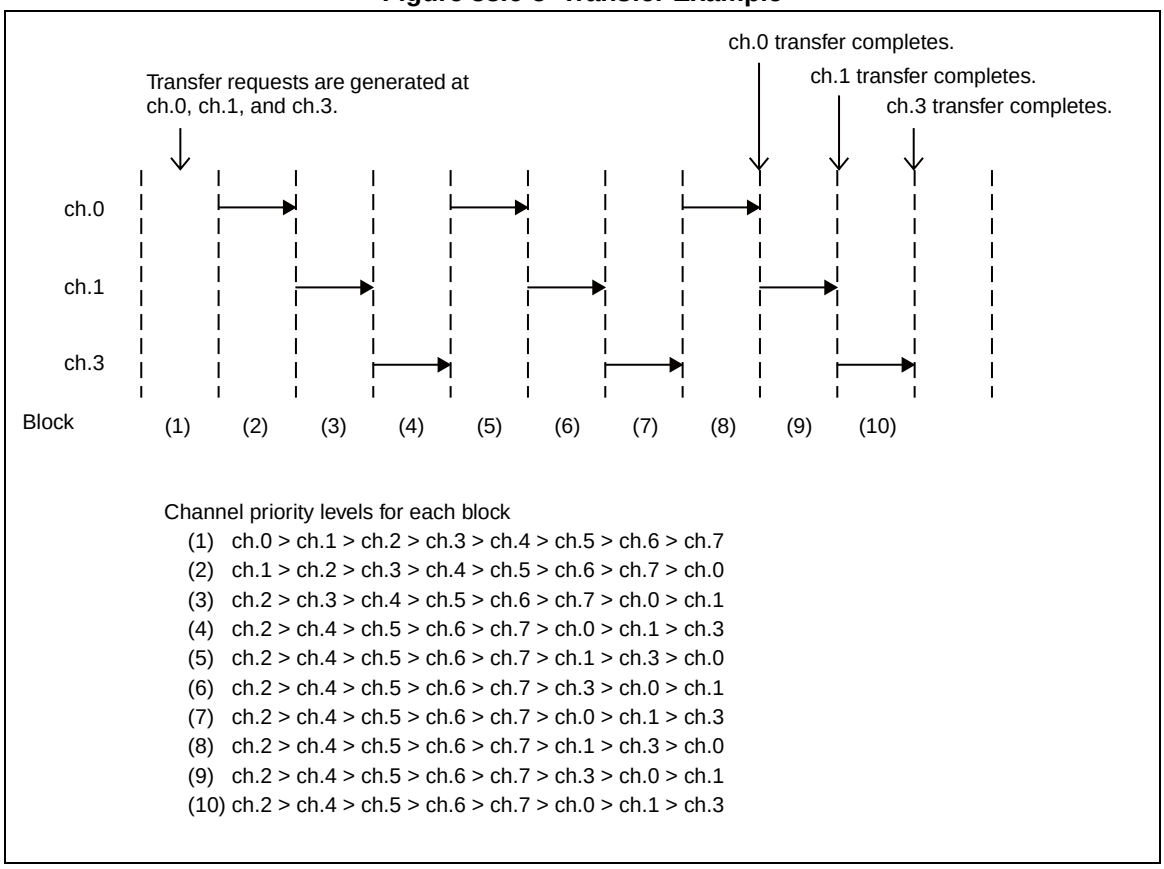
  Figure 33.6-5 shows an example of transfer under the following conditions.

  Transfer request: Occurs simultaneously at ch.0, ch.1, and ch.3.

  Transfer mode: Burst transfer mode in all channels

  Transfer count: 3 for all channels.

Figure 33.6-5 Transfer Example



Channel priority levels for each block

(1) ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7
(2) ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0
(3) ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1
(4) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3
(5) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.1 > ch.3 > ch.0
(6) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.3 > ch.0 > ch.1
(7) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3
(8) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.1 > ch.3 > ch.0
(9) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.3 > ch.0 > ch.1
(10) ch.2 > ch.4 > ch.5 > ch.6 > ch.7 > ch.0 > ch.1 > ch.3

1. Transfer requests are generated simultaneously at ch.0, ch.1, and ch.3.
2. 1 block of data at ch.0 is transferred.
3. After 1 block of data at ch.0 has been transferred, 1 block of data at ch.1 is transferred.
4. After 1 block of data at ch.1 has been transferred, 1 block of data at ch.3 is transferred.
5. After 1 block of data at ch.3 has been transferred, the second block of data at ch.0 is transferred.
6. After the second block of data at ch.0 has been transferred, the second block of data at ch.1 is transferred.
7. After the second block of data at ch.1 has been transferred, the second block of data at ch.3 is transferred.
8. After the second block of data at ch.3 has been transferred, the third block of data at ch.0 is transferred.
   Transfer at ch.0 ends.
9. After the third block of data at ch.0 has been transferred, the third block of data at ch.1 is transferred.
   Transfer at ch.1 ends.
10. After the third block of data at ch.1 has been transferred, the third block of data at ch.3 is transferred.
    Transfer at ch.3 ends.

■ Transfer Address Update Operation

Upon every completion of data transfer of the data size specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7), the transfer source address and transfer destination address can be incremented/decremented.

Address updates can be set with the following registers.

- Transfer source address: SAC1 and SAC0 bits of DMA channel control registers (DCCR0 to DCCR7)
- Transfer destination address: DAC1 and DAC0 bits of DMA channel control registers (DCCR0 to DCCR7)

The increment/decrement width varies depending on the size specified by the TS1 and TS0 bits of DMA channel control registers (DCCR0 to DCCR7).

Table 33.6-3 shows the relationship between the bit setting value and the address increment/decrement width.

Table 33.6-3 Bit Setting Value and Increment/Decrement Width

Transfer Source Address (SAC1 And SAC0)	Transfer Destination Address (DAC1 And DAC0)	Transfer Size (TS1 And TS0)	Transfer Source Address Increment/ Decrement Width	Transfer Destination Address Increment/ Decrement Width
00 (increment)	00 (increment)	00 (8 bits)	Incremented by 1	Incremented by 1
		01 (16 bits)	Incremented by 2	Incremented by 2
		10 (32 bits)	Incremented by 4	Incremented by 4
	01 (decrement)	00 (8 bits)	Incremented by 1	Decremented by 1
		01 (16 bits)	Incremented by 2	Decremented by 2
		10 (32 bits)	Incremented by 4	Decremented by 4
	11 (fixed)	00 (8 bits)	Incremented by 1	No increase or decrease
		01 (16 bits)	Incremented by 2	No increase or decrease
		10 (32 bits)	Incremented by 4	No increase or decrease
01 (decrement)	00 (increment)	00 (8 bits)	Decremented by 1	Incremented by 1
		01 (16 bits)	Decremented by 2	Incremented by 2
		10 (32 bits)	Decremented by 4	Incremented by 4
	01 (decrement)	00 (8 bits)	Decremented by 1	Decremented by 1
		01 (16 bits)	Decremented by 2	Decremented by 2
		10 (32 bits)	Decremented by 4	Decremented by 4
	11 (fixed)	00 (8 bits)	Decremented by 1	No increase or decrease
		01 (16 bits)	Decremented by 2	No increase or decrease
		10 (32 bits)	Decremented by 4	No increase or decrease
11 (fixed)	00 (increment)	00 (8 bits)	No increase or decrease	Incremented by 1
		01 (16 bits)	No increase or decrease	Incremented by 2
		10 (32 bits)	No increase or decrease	Incremented by 4
	01 (decrement)	00 (8 bits)	No increase or decrease	Decremented by 1
		01 (16 bits)	No increase or decrease	Decremented by 2
		10 (32 bits)	No increase or decrease	Decremented by 4
	11 (fixed)	00 (8 bits)	No increase or decrease	No increase or decrease
		01 (16 bits)	No increase or decrease	No increase or decrease
		10 (32 bits)	No increase or decrease	No increase or decrease

■ Transfer Request Acceptance/Transfer Complete Signal Output

● Output timing

If transfer source type/transfer destination type is set to other than memory (such as peripheral functions) by the ST bit or DT bit (ST/DT = 1) of DMA channel control registers (DCCR0 to DCCR7), transfer request acceptance/transfer complete signal can be output.

- The ST bit is "1":
  - The transfer request acceptance signal is output for the transfer request in the reading cycle of the transfer source (every time).
  - During the last transfer specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer complete signal is output in the reading cycle of the transfer source (only once).
- The DT bit is "1":
  - The transfer request acceptance signal is output for the transfer request in the writing cycle of the transfer destination (every time).
  - During the last transfer specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer complete signal is output in the writing cycle of the transfer destination (only once).

<Notes>

- The transfer request acceptance/transfer complete signal that have been output can be verified only when the transfer source/transfer destination is set to the external bus interface.

  For details, see "CHAPTER 13 External Bus Interface".

  If the transfer source/transfer destination is set to other than the external bus interface, the transfer request acceptance/transfer complete signal cannot be verified.
- If the transfer mode is set to demand transfer by the TM1 and TM0 bits (TM1, TM0 = 11) of DMA channel control registers (DCCR0 to DCCR7), be sure to set at least one of the following bits to "1".
  - ST bit of DMA channel control registers (DCCR0 to DCCR7)
  - DT bit of DMA channel control registers (DCCR0 to DCCR7)

● Note

Transfer wait may be generated depending on the ST bit/DT bit setting of the DMA channel control registers (DCCR0 to DCCR7).

- Both the ST bit and DT bit at ch.0 are set to "1":

  Transfer wait is generated between the points from the transfer source to the transfer destination or between the points from transfer destination to the transfer source during the transfer at ch.0.
- Transition from the channel in which the transfer destination type is specified to the channel in which transfer source type is specified:

  Setting examples: Transition of transfer channel from ch.0 to ch.1

Channel Number	ST	DT
ch.0	0	1
ch.1	1	0

Transfer wait is generated between the transfer destination of the first transfer channel (ch.0) and the transfer source of the second transfer channel (ch.1).

33.6.3 Transfer Suspension

DMA controller (DMAC) suspends the DMA transfer under the following cases.
This section explains the operation when DMA transfer is suspended.

■ Overview

DMA transfer is suspended when:

- "0" is written to the DME bit of DMA control registers (DMACR).
- "0" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7).
- Transfer stop request is output from the transfer request source.

■ Transfer Suspension/Restart

Transfer is suspended in units of blocks. For this reason, transfer is suspended after the completion of the transfer of 1 block of data if any suspension sources are generated during the transfer.

Once a transfer is suspended, DMA controller (DMAC) enters the stop state without executing new transfer.

- **"0" is written to the DME bit of DMA control registers (DMACR)**

  All channels enter the stop state.

  When the DME bit is cleared to "0", the channel where the transfer is being executed suspends the transfer on the completion of the transfer of 1 block of the data being transferred. Also, transfer requests that have been detected are not cleared.

  Restart the DMA transfer by using the following procedure.

  1. Write "1" to the DME bit of DMA control registers (DMACR).

- **"0" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7)**

  The corresponding channels enter the stop state.

  If data is being transferred at the corresponding channel, the transfer is suspended upon the completion of the transfer of 1 block of the data currently being transferred. Also, transfer requests that have been detected are cleared.

  Restart the DMA transfer by using the following procedure.

  1. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7) of which channels are in the stop state.
  2. Generate a new transfer request.

- **Suspension due to a transfer stop request from transfer request source**

  If a transfer error occurs and transfer stop request is generated when the DMA controller (DMAC) is activated in multifunction serial interface, the transfer is suspended upon the completion of the transfer of 1 block of the data currently being transferred.

  Once the transfer is suspended, the following states are generated.

  - The SP bit of DMA channel status registers (DCSR0 to DCSR7) is changed to "1".

- The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".
- Transfer requests that have been detected are cleared.

While a transfer stop request is being output, no new transfer request is accepted.

Restart the DMA transfer by using the following procedure.

1. Disable the transfer stop request.
2. Write "0" to the SP bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel.
3. Write "1" to the CE bit of the DMA channel control registers (DCCR0 to DCCR7).
4. Generate a new transfer request.

<Note>

The SP bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the SP bit.

■ Restart Operation

Follow the restart procedures to resume DMA transfer. The restart operations vary when "1" is written to the DME bit of DMA control register (DMACR) and when "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7).

Also, they vary depending on the transfer mode specified.

Table 33.6-4 shows the operation when the transfer is restarted.

Table 33.6-4 Operation of Transfer Restart

Transfer Mode	"1" Is Written to the DME Bit.	"1" Is Written to the CE Bit.
Block transfer	If a new transfer request is detected, the transfer restarts in accordance with its priority.	If a new transfer request is detected, the transfer restarts in accordance with its priority. (To generate a new transfer request in demand transfer mode, a DMA transfer request must be input once again from the DREQ0 to DREQ3 pins.)
Burst transfer	Transfer restarts immediately in accordance with the priority.	
Demand transfer	If transfer requests are generated continuously when "1" is written to the DME bit, transfer restarts immediately without determining the priority.	

<Note>

For details on input timing of DMA transfer requests from the DREQ0 to DREQ3 pins, see "CHAPTER 13 External Bus Interface".

33.6.4 Operation at the End of Transfer

This section explains the end operation of DMA transfer.

The end of transfer can be classified as normal end or abnormal end.

- Normal end

  DMA transfer ends normally when the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7) have completed.

  When DMA transfer ends normally, the following conditions occur.

  1. The NC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1".
  2. The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".

     DMA controller (DMAC) enters the stop state.

  However, if the transfer request source is set to other than software and reload of transfer count is set, the CE bit of DMA channel control registers (DCCR0 to DCCR7) is not cleared.

  Also, if "1" is written to the CE bit of DMA channel control registers (DCCR0 to DCCR7) for the corresponding channel when the value set for DMA transfer count registers (DTCR0 to DTCR7) is "0", the NC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1" as with the normal end.

  Be sure to set "1" or greater as the value of DMA transfer count registers (DTCR0 to DTCR7) before writing "1" to the CE bit of DMA channel control registers (DCCR0 to DCCR7).

<Notes>

- If interrupt requests are selected for the transfer request source, the interrupt request flags for peripheral functions are cleared by the DMA controller (DMAC) once DMA transfer has completed.
- The NC bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the NC bit.

- Abnormal end

  An abnormal end interrupt request is output when the prohibited value is set to DMA channel control registers (DCCR0 to DCCR7).

  If the value of DMA channel control registers (DCCR0 to DCCR7) is any of the following, the DMA transfer is regarded as having an abnormal end.

  - TM1, TM0 bits = 10 (setting prohibited)
  - SAC1, SAC0 bits = 10 (setting prohibited)
  - DAC1, DAC0 bits = 10 (setting prohibited)
  - TS1, TS0 bits = 11 (setting prohibited)
  - RS1, RS0 bits = 00 and TM1, TM0 bits = 11 (transfer request source: software, transfer mode: demand transfer)

  When DMA transfer ends abnormally, the following conditions occur.

  1. The AC bit of DMA channel status registers (DCSR0 to DCSR7) for the corresponding channel is changed to "1".
  2. The CE bit of DMA channel control registers (DCCR0 to DCCR7) is changed to "0".

     DMA controller (DMAC) enters the stop state.

<Note>

The AC bit of DMA channel status registers (DCSR0 to DCSR7) is not cleared to "0" automatically. To clear this bit, write "0" to the AC bit.

33.6.5 Post-transfer Operation

This section explains the post-DMA-transfer operation of the blocks with the specified transfer count.

■ Reload Operation

DMA controller (DMAC) is equipped with the reload register that maintains the transfer source address, transfer destination address, and transfer count, which have been written prior to the transfer. With this function, the transfer source address, transfer destination address, and transfer count that have been specified prior to the transfer can be reloaded after the transfer.

The following registers provide reload registers.

- DMA source address registers (DSAR0 to DSAR7)
- DMA destination address registers (DDAR0 to DDAR7)
- DMA transfer count registers (DTCR0 to DTCR7)

<Note>

The reload register is used as a register that memorizes the value written to each corresponding register. The value of the reload register cannot be read.

● Reload Operation of Transfer Source Address

To specify whether to reload the transfer source address to DMA source address registers (DSAR0 to DSAR7), use the SAR bit of DMA channel control registers (DCCR0 to DCCR7).

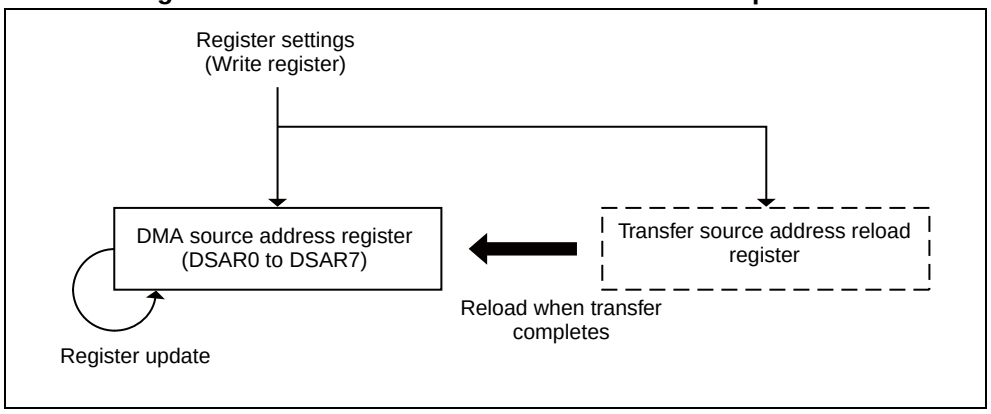
Table 33.6-5 shows the setting method and post-transfer operation.

Table 33.6-5 Setting Method and Post-transfer Operation

SAR	Post-transfer Operation
0	The value of the DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	After transfer completion, the value of DMA source address registers (DSAR0 to DSAR7) returns to the value that is written prior to the transfer.

Figure 33.6-6 shows the operation when reload is enabled by the SAR bit (SAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 33.6-6 Transfer Source Address Reload Operation



<Note>

If a transfer is suspended or abnormal end occurs before completing the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer source address is not reloaded even when the reload is enabled by the SAR bit (SAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

The value of DMA source address registers (DSAR0 to DSAR7) becomes the next address subsequent to the last-accessed address.

● Reload Operation of Transfer Destination Address

To specify whether to reload the transfer destination address to DMA destination address registers (DDAR0 to DDAR7), use the DAR bit of DMA channel control registers (DCCR0 to DCCR7).

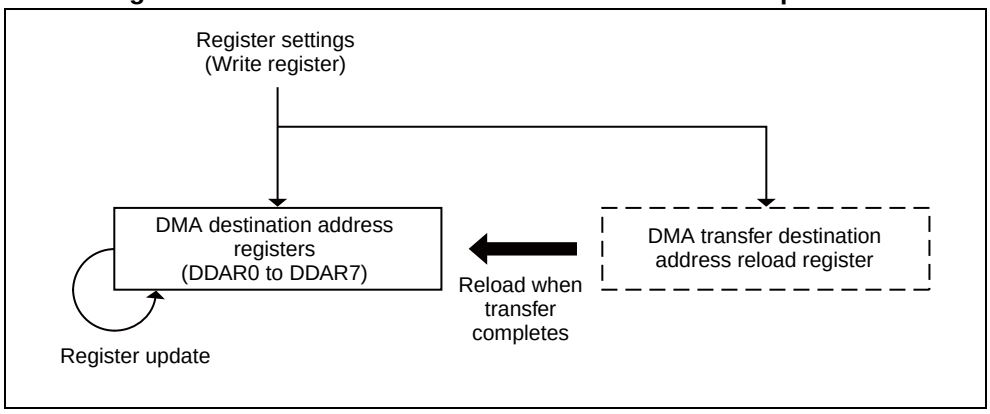
Table 33.6-6 shows the setting method and post-transfer operation.

Table 33.6-6 Setting Method and Post-transfer Operation

DAR	Post-transfer Operation
0	The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address after transfer completion.
1	After transfer completion, the value of DMA destination address registers (DDAR0 to DDAR7) returns to the value that is written prior to the transfer.

Figure 33.6-7 shows the operation when reload is enabled by the DAR bit (DAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 33.6-7 Transfer Destination Address Reload Operation



<Note>

If a transfer is suspended or abnormal end occurs before completing the data transfer for the number of times specified at DMA transfer count registers (DTCR0 to DTCR7), the transfer destination address is not reloaded even when the reload is enabled by the DAR bit (DAR = 1) of DMA channel control registers (DCCR0 to DCCR7).

The value of DMA destination address registers (DDAR0 to DDAR7) becomes the next address subsequent to the last-accessed address.

● Reload operation of transfer count

To specify whether to reload the transfer count to DMA transfer count registers (DTCR0 to DTCR7), use the TCR bit of DMA channel control registers (DCCR0 to DCCR7).

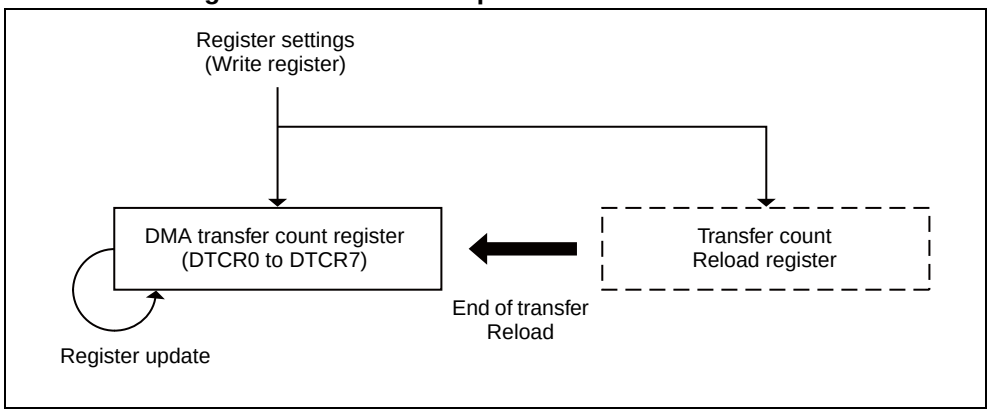
Table 33.6-7 shows the setting method and post-transfer operation.

Table 33.6-7 Setting Method and Post-transfer Operation

TCR	Post-transfer Operation
0	After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) is cleared to "0".
1	After transfer completion, the value of DMA transfer count registers (DTCR0 to DTCR7) returns to the value that is written prior to the transfer.

Figure 33.6-8 shows the operation when reload is enabled by the TCR bit (TCR = 1) of DMA channel control registers (DCCR0 to DCCR7).

Figure 33.6-8 Reload Operation of Transfer Count



Also, whether to clear the CE bit of DMA channel control registers (DCCR0 to DCCR7) upon the completion of the transfer depends on the TCR bit setting of the DMA channel control registers (DCCR0 to DCCR7).

Table 33.6-8 shows the relationship between the TCR bit and CE bit after transfer completion.

Table 33.6-8 Relationship between the TCR Bit and the Post-transfer CE Bit

TCR	Transfer Request Source	
	Software	Other Than Software
0	The CE bit is cleared to "0".	The CE bit is cleared to "0".
1	The CE bit is cleared to "0".	The CE bit is not cleared.

<Notes>

- If the reload of the transfer count is enabled by the TCR bit (TCR = 1) of DMA channel control registers (DCCR0 to DCCR7), transfer continues sequentially in demand transfer mode while transfer requests are being output.
- If the DMA transfer is suspended or abnormal end occurs, DMA transfer count registers (DTCR0 to DTCR7) show the number of remaining transfers.

33.6.6 DMA Transfer Halt

If an interrupt request is generated from the a peripheral function, DMA transfer is halted.

DMA transfer is halted in units of blocks. Therefore, if an interrupt request of a level that is higher than the interrupt level specified at DMA-halt by interrupt level register (DILVR) is generated, the transfer is halted after the completion of the transfer of 1 block of the data currently being transferred.

If the DMA transfer is halted, DMA controller (DMAC) enters the halt state without executing new transfer.

The DMA transfer restarts when the interrupt level reaches or falls below the level of LVL4 to LVL0 bits of DMA-halt by interrupt level register (DILVR).

Table 33.6-9 shows the relationship between the LVL4 to LVL0 bit settings of DMA-halt by interrupt level register (DILVR) and the interrupt request level that halts the DMA transfer.

Table 33.6-9 Interrupt Request Level where DMA Transfers are Halted.

LVL4	LVL3	LVL2	LVL1	LVL0	Interrupt Request Level Where DMA Transfers Are Halted.
1	0	0	0	0	DMA transfers are not halted.
1	0	0	0	1	Interrupt request of "11_H" or higher
1	0	0	1	0	Interrupt request of "12_H" or higher
1	0	0	1	1	Interrupt request of "13_H" or higher
1	0	1	0	0	Interrupt request of "14_H" or higher
1	0	1	0	1	Interrupt request of "15_H" or higher
1	0	1	1	0	Interrupt request of "16_H" or higher
1	0	1	1	1	Interrupt request of "17_H" or higher
1	1	0	0	0	Interrupt request of "18_H" or higher
1	1	0	0	1	Interrupt request of "19_H" or higher
1	1	0	1	0	Interrupt request of "$1A_H$" or higher
1	1	0	1	1	Interrupt request of "$1B_H$" or higher
1	1	1	0	0	Interrupt request of "$1C_H$" or higher
1	1	1	0	1	Interrupt request of "$1D_H$" or higher
1	1	1	1	0	Interrupt request of "$1E_H$" or higher
1	1	1	1	1	All interrupt requests

CHAPTER 34 Select Function for DMA Transfer Request Generation/Clear by a Peripheral Function

This section explains the method of generating a DMA transfer request by using an interrupt request of peripheral functions and the method of clearing an interrupt request flag of peripheral functions from the DMA controller (DMAC).

34.1 Overview

34.2 Configuration

34.3 Registers

34.4 An Explanation of Operations and Setting Procedure Examples

34.1 Overview

This series enables the activation of a DMA transfer by using an interrupt request of peripheral functions.
Registers for selecting an interrupt request which activates a DMA transfer are provided for each channel of the DMA controller (DMAC).
If multiple interrupt requests are allocated to one interrupt vector number, you must also set which interrupt request flag is to be cleared using the DMA controller (DMAC).

■ Overview of Generation of a DMA Transfer Request by Using a Peripheral Function

The registers of the DMA controller (DMAC) can be used to set the source that triggers a DMA transfer request generation (transfer request source) for an interrupt request of a peripheral function.

Values corresponding to the interrupt vector number are specified to select the interrupt request to be used.

■ Overview of the Select Function for DMA Transfer Request Clear by Using a Peripheral Function

- Selection of an interrupt request

  If the source that triggers a DMA transfer request generation (transfer request source) is designated as an interrupt request of a peripheral function, the interrupt request flag is cleared by the DMA controller (DMAC) after the DMA transfer.

  For this reason, if multiple interrupt requests are allocated to the interrupt vector number to be selected as the source that triggers a DMA transfer request generation (transfer request source), you must select the interrupt request flag to be cleared by the DMA controller (DMAC) after the DMA transfer.

- Selection of a transfer stop request

  In the UART/CSIO/I^2C ch.8 to ch.11, if an interrupt request is generated upon reception, a transfer stop request is output to the DMA controller (DMAC) to interrupt the DMA transfer.

  Use this function to select the channel of which reception interrupt request is to be used as the DMA transfer stop request.

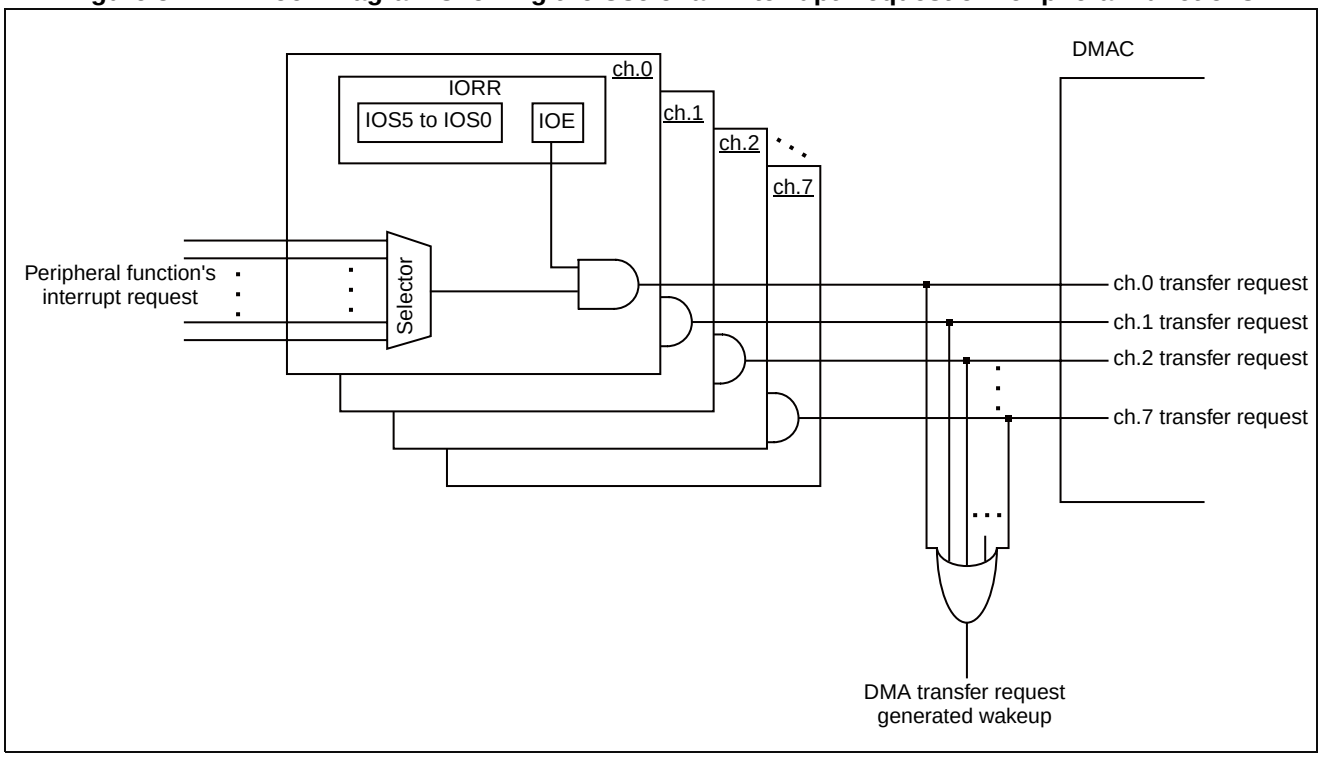
34.2 Configuration

This section explains the part at which a DMA transfer request is generated by a peripheral function and the configuration of the select function for DMA transfer request clear.

■ Block Diagram of the Part at which a DMA Transfer Request is Generated by a Peripheral Function

Figure 34.2-1 is a block diagram of the part which uses an interrupt request of a peripheral function as a transfer request source for the DMA transfer.

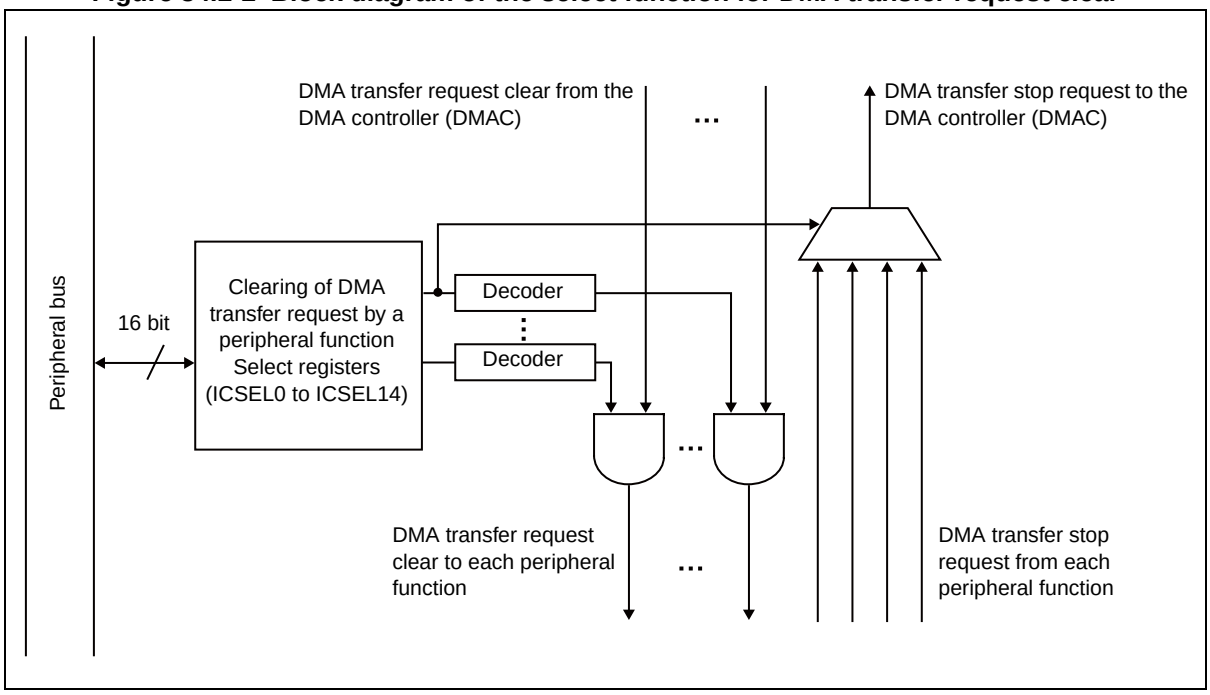
Figure 34.2-1 Block Diagram showing the Use of an Interrupt Request of Peripheral Functions



■ Block diagram of the select function for DMA transfer request clear

Figure 34.2-2 is a block diagram of the select function for DMA transfer request clear.

Figure 34.2-2 Block diagram of the select function for DMA transfer request clear


DMA transfer request clear from the DMA controller (DMAC)
DMA transfer stop request to the DMA controller (DMAC)
Peripheral bus
16 bit
Clearing of DMA transfer request by a peripheral function Select registers (ICSEL0 to ICSEL14)
Decoder
Decoder
DMA transfer request clear to each peripheral function
DMA transfer stop request from each peripheral function

- Select Register for DMA transfer request clear by a peripheral function (ICSEL0 to ICSEL14)
  This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).
- Decoder

34.3 Registers

This section explains the configurations and functions of registers of the select function for DMA transfer request generation/clear.

■ List of registers of the part at which a DMA transfer request is generated by a peripheral function

Table 34.3-1 is a list of the registers of the part at which a DMA transfer request is generated.

Table 34.3-1 Registers of the part at which a DMA transfer request is generated by a peripheral function

Channel of DMAC	Abbreviated Register Name	Register Name	Reference
0	IORR0	IO-data request register 0	34.3.1
1	IORR1	IO-data request register 1	34.3.1
2	IORR2	IO-data request register 2	34.3.1
3	IORR3	IO-data request register 3	34.3.1
4	IORR4	IO-data request register 4	34.3.1
5	IORR5	IO-data request register 5	34.3.1
6	IORR6	IO-data request register 6	34.3.1
7	IORR7	IO-data request register 7	34.3.1

■ List of registers of the select function for DMA transfer request clear

Table 34.3-2 shows a list of the registers of the select function for DMA transfer request clear.

Table 34.3-2 List of registers of the select function for DMA transfer request clear

Channel	Abbreviated Register Name	Register Name	Reference
Common	ICSEL0	Select register 0 for DMA transfer request clear by a peripheral function	34.3.2
	ICSEL1	Select register 1 for DMA transfer request clear by a peripheral function	34.3.3
	ICSEL2	Select register 2 for DMA transfer request clear by a peripheral function	34.3.4
	ICSEL3	Select register 3 for DMA transfer request clear by a peripheral function	34.3.5
	ICSEL4	Select register 4 for DMA transfer request clear by a peripheral function	34.3.6
	ICSEL5	Select register 5 for DMA transfer request clear by a peripheral function	34.3.7
	ICSEL6	Select register 6 for DMA transfer request clear by a peripheral function	34.3.8
	ICSEL7	Select register 7 for DMA transfer request clear by a peripheral function	34.3.9
	ICSEL8	Select register 8 for DMA transfer request clear by a peripheral function	34.3.10
	ICSEL9	Select register 9 for DMA transfer request clear by a peripheral function	34.3.11
	ICSEL10	Select register 10 for DMA transfer request clear by a peripheral function	34.3.12
	ICSEL11	Select register 11 for DMA transfer request clear by a peripheral function	34.3.13
	ICSEL12	Select register 12 for DMA transfer request clear by a peripheral function	34.3.14
	ICSEL13	Select register 13 for DMA transfer request clear by a peripheral function	34.3.15
	ICSEL14	Select register 14 for DMA transfer request clear by a peripheral function	34.3.16

34.3.1 IO-Data Request Registers (IORR0 to IORR7)

This register sets which interrupt request of peripheral functions is to be the source that triggers a DMA transfer request generation when the source is set as an interrupt request of a peripheral function.
This register is provided for each channel of the DMA controller (DMAC).

Figure 34.3-1 shows the bit configuration of the IO-data request registers (IORR0 to IORR7).

Figure 34.3-1 Bit configuration of the IO-data request registers (IORR0 to IORR7)

bit	7	6	5	4	3	2	1	0
	Reserved	IOE	IOS5	IOS4	IOS3	IOS2	IOS1	IOS0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

This register becomes enabled when the source that triggers a DMA transfer request generation is set as an interrupt request of a peripheral function (RS1, RS0 = 01) in the RS1 and RS0 bit of the DMA channel control registers (DCCR0 to DCCR7).

[bit7]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit6]: IOE (transfer request enable bit)

This bit sets whether a DMA transfer request is output to the DMA controller (DMAC) of the corresponding channel when an interrupt request specified in bits ranging from IOS5 to IOS0 is generated.

Written Value	Explanation
0	A DMA transfer request is not output. (An interrupt request from a peripheral function is not used as a DMA transfer request.)
1	A DMA transfer request is output. (An interrupt request from a peripheral function is used as a DMA transfer request.)

[bit5 to bit0]: IOS5 to IOS0 (transfer request selection bit)

These bits set which interrupt request generated from a peripheral function is to be used as a transfer request source by the channel of the DMA controller (DMAC) corresponding to this register.

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
000000	16	10	External interrupt request ch.0 to ch.7
000001	17	11	External interrupt request ch.8 to ch.15
000010	18	12	External interrupt request ch.16 to ch.23
000011	19	13	External interrupt request ch.24 to ch.31
000100	20	14	16-bit reload timer ch.0 to ch.2
000101	21	15	Reception interrupt request from UART/CSIO ch.0
000110	22	16	Transmission interrupt request from UART/CSIO ch.0 Transmission bus idle interrupt request from UART/CSIO ch.0
000111	23	17	Reception interrupt request from UART/CSIO/I^2C ch.1
001000	24	18	Transmission interrupt request from UART/CSIO/I^2C ch.1 Transmission bus idle interrupt request from UART/CSIO ch.1
001001	25	19	-
001010	26	1A	Reception interrupt request from UART/CSIO/I^2C ch.2
001011	27	1B	Transmission interrupt request from UART/CSIO/I^2C ch.2 Transmission bus idle interrupt request from UART/CSIO ch.2
001100	28	1C	-
001101	29	1D	Reception interrupt request from UART/CSIO/I^2C ch.3
001110	30	1E	Transmission interrupt request from UART/CSIO/I^2C ch.3 Transmission bus idle interrupt request from UART/CSIO ch.3

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
001111	31	1F	Reception interrupt request from UART/CSIO/I^2C ch.4
010000	32	20	Transmission interrupt request from UART/CSIO/I^2C ch.4 Transmission bus idle interrupt request from UART/CSIO ch.4
010001	33	21	Reception interrupt request from UART/CSIO/I^2C ch.5
010010	34	22	Transmission interrupt request from UART/CSIO/I^2C ch.5 Transmission bus idle interrupt request from UART/CSIO ch.5
010011	35	23	Reception interrupt request from UART/CSIO/I^2C ch.6
010100	36	24	Transmission interrupt request from UART/CSIO/I^2C ch.6 Transmission bus idle interrupt request from UART/CSIO/I^2C ch.6
010101	37	25	Reception interrupt request from UART/CSIO/I^2C ch.7 Edge detection interrupt request from 32-bit input capture ch.4 to ch.7
010110	38	26	Transmission interrupt request from UART/CSIO/I^2C ch.7 Transmission bus idle interrupt request from UART/CSIO ch.7
010111	39	27	Reception interrupt request from UART/CSIO/I^2C ch.8 to ch.11 Transmission interrupt request from UART/CSIO/I^2C ch.8 to ch.11 Transmission bus idle interrupt request from UART/CSIO ch.8 to ch.11 Transmission FIFO interrupt request from UART/CSIO/I^2C ch.8 to ch.11
011000	40	28	-
011001	41	29	Main timer interrupt request Sub timer interrupt request Underflow interrupt request from the watch counter
011010	42	2A	10-bit A/D converter - A/D scan conversion interrupt request - A/D priority conversion interrupt request
011011	43	2B	-
011100	44	2C	Edge detection interrupt request from 32-bit input capture ch.0 to ch.3
011101	45	2D	-

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
011110	46	2E	Base timer ch.0 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
011111	47	2F	Base timer ch.1 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100000	48	30	Base timer ch.2 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100001	49	31	Base timer ch.3 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100010	50	32	Base timer ch.4, ch.5 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100011	51	33	Base timer ch.6, ch.7 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100100	52	34	Base timer ch.8, ch.9 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request

IOS5 to IOS0	Interrupt Vector Number		Peripheral Function
	Decimal	Hexadecimal	
100101	53	35	Base timer ch.10, ch.11 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100110	54	36	Base timer ch.12 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
100111	55	37	Base timer ch.13 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
101000	56	38	Base timer ch.14, ch.15 - Underflow interrupt request - Overflow interrupt request - Duty match interrupt request - Trigger interrupt request - Measurement end interrupt request
101001	57	39	-
101010	58	3A	-
101011	59	3B	-
101100	60	3C	-
101101	61	3D	-
101110	62	3E	-
101111	63	3F	-

<Notes>

- If one interrupt vector number is used by multiple interrupt requests, only one interrupt request can be used as a DMA transfer request source.

  Disable the generation of an interrupt request which is not designated as a DMA transfer request source.
- If one interrupt vector number is used by multiple interrupt requests, set an interrupt request to clear the flag bit in the select register for DMA transfer request clear by a peripheral function (ICSEL0 to ICSEL14).
- Set an interrupt level for the interrupt request selected in this register so that values in the interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) indicate the following values.

  ILM ≤ ICR

34.3.2 Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)

The external interrupt request ch.0 to ch.7 is assigned to interrupt vector number 16 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-2 shows the bit configuration of select register 0 for DMA transfer request clear by a peripheral function (ICSEL0).

Figure 34.3-2 Bit Configuration of Select Register 0 for DMA Transfer Request Clear by a Peripheral Function (ICSEL0)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL02	EISEL01	EISEL00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: EISEL02 to EISEL00 (interrupt request select bit)

These bits select the flag bit to be cleared in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 16 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 16 (decimal), the bit selected here will be cleared.

EISEL02	EISEL01	EISEL00	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.0	EIRR0: ER0
0	0	1	External interrupt request ch.1	EIRR0: ER1
0	1	0	External interrupt request ch.2	EIRR0: ER2
0	1	1	External interrupt request ch.3	EIRR0: ER3
1	0	0	External interrupt request ch.4	EIRR0: ER4
1	0	1	External interrupt request ch.5	EIRR0: ER5
1	1	0	External interrupt request ch.6	EIRR0: ER6
1	1	1	External interrupt request ch.7	EIRR0: ER7

34.3.3 Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)

The external interrupt request ch.8 to ch.15 is assigned to interrupt vector number 17 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-3 shows the bit configuration of select register 1 for DMA transfer request clear by a peripheral function (ICSEL1).

Figure 34.3-3 The Bit Configuration of Select Register 1 for DMA Transfer Request Clear by a Peripheral Function (ICSEL1)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL12	EISEL11	EISEL10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: EISEL12 to EISEL10 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 17 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 17 (decimal), the bit selected here will be cleared.

EISEL12	EISEL11	EISEL10	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.8	EIRR1: ER8
0	0	1	External interrupt request ch.9	EIRR1: ER9
0	1	0	External interrupt request ch.10	EIRR1: ER10
0	1	1	External interrupt request ch.11	EIRR1: ER11
1	0	0	External interrupt request ch.12	EIRR1: ER12
1	0	1	External interrupt request ch.13	EIRR1: ER13
1	1	0	External interrupt request ch.14	EIRR1: ER14
1	1	1	External interrupt request ch.15	EIRR1: ER15

34.3.4 Select Register 2 for DMA Transfer Request Clear by a Peripheral Function (ICSEL2)

The external interrupt request ch.16 to ch.23 is assigned to interrupt vector number 18 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-4 shows the bit configuration of select register 2 for a DMA transfer request clear by a peripheral function (ICSEL2).

Figure 34.3-4 Bit Configuration of Select Register 2 for a DMA Transfer Request Clear by a Peripheral Function (ICSEL2)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL22	EISEL21	EISEL20
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: EISEL22 to EISEL20 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 18 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 18 (decimal), the bit selected here will be cleared.

EISEL22	EISEL21	EISEL20	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.16	EIRR2: ER16
0	0	1	External interrupt request ch.17	EIRR2: ER17
0	1	0	External interrupt request ch.18	EIRR2: ER18
0	1	1	External interrupt request ch.19	EIRR2: ER19
1	0	0	External interrupt request ch.20	EIRR2: ER20
1	0	1	External interrupt request ch.21	EIRR2: ER21
1	1	0	External interrupt request ch.22	EIRR2: ER22
1	1	1	External interrupt request ch.23	EIRR2: ER23

34.3.5 Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3)

The external interrupt request ch.24 to ch.31 is assigned to interrupt vector number 19 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-5 shows the bit configuration of select register 3 for DMA transfer request clear by a peripheral function (ICSEL3).

Figure 34.3-5 The Bit Configuration of Select Register 3 for DMA Transfer Request Clear by a Peripheral Function (ICSEL3)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	EISEL32	EISEL31	EISEL30
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: EISEL32 to EISEL30 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 19 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 19 (decimal), the bit selected here will be cleared.

EISEL32	EISEL31	EISEL30	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	External interrupt request ch.24	EIRR3: ER24
0	0	1	External interrupt request ch.25	EIRR3: ER25
0	1	0	External interrupt request ch.26	EIRR3: ER26
0	1	1	External interrupt request ch.27	EIRR3: ER27
1	0	0	External interrupt request ch.28	EIRR3: ER28
1	0	1	External interrupt request ch.29	EIRR3: ER29
1	1	0	External interrupt request ch.30	EIRR3: ER30
1	1	1	External interrupt request ch.31	EIRR3: ER31

34.3.6 Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4)

The interrupt request of the 16-bit reload timer ch.0 to ch.2 is assigned to interrupt vector number 20 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-6 shows the bit configuration of select register 4 for DMA transfer request clear by a peripheral function (ICSEL4).

Figure 34.3-6 The Bit Configuration of Select Register 4 for DMA Transfer Request Clear by a Peripheral Function (ICSEL4)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RTSEL1	RTSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit1, bit0]: RTSEL1 to RTSEL0 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 20 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 20 (decimal), the bit selected here will be cleared.

RTSEL1	RTSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Underflow interrupt request of the 16-bit reload timer ch.0	TMCSR0: UF
0	1	Underflow interrupt request of the 16-bit reload timer ch.1	TMCSR1: UF
1	0	Underflow interrupt request of the 16-bit reload timer ch.2	TMCSR2: UF
1	1		

34.3.7 Select Register 5 for DMA Transfer Request Clear by a Peripheral Function (ICSEL5)

The interrupt request of ch.8 to ch.11 of the UART/CSIO/I^2C is assigned to interrupt vector number 39 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).
In addition, this register also selects a channel to be used to output a DMA transfer stop request.

Figure 34.3-7 shows the bit configuration of select register 5 for DMA transfer request clear by a peripheral function (ICSEL5).

Figure 34.3-7 The Bit Configuration of Select Register 5 for DMA Transfer Request Clear by a Peripheral Function (ICSEL5)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	MFSSEL2	MFSSEL1	MFSSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: MFSSEL2 to MFSSEL0 (interrupt request select bit)

These bits select the following two items.

1. These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 39 (decimal).

   When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 39 (decimal), the flag bit of the interrupt request selected with these bits will be cleared.

2. These bits select a channel to output a transfer stop request to the DMA controller (DMAC).

   When a reception interrupt request is generated in the channel selected with these bits, a DMA transfer stop request is output to the DMA controller (DMAC).

MFSSEL2	MFSSEL1	MFSSEL0	Interrupt Request To Be Cleared by the DMAC		Channel To Be Used To Output a DMA Transfer Stop Request
			Interrupt Request Name	Flag Bit To Be Cleared	
0	0	0	Reception interrupt request from UART/CSIO/I^{2}C ch.8	SSR8: RDRF SSR8: ORE SSR8: FRE SSR8: PE	Reception interrupt request from UART/ CSIO/I^{2}C ch.8
0	0	1	Reception interrupt request from UART/CSIO/I^{2}C ch.9	SSR9: RDRF SSR9: ORE SSR9: FRE SSR9: PE	Reception interrupt request from UART/ CSIO/I^{2}C ch.9
0	1	0	Reception interrupt request from UART/CSIO/I^{2}C ch.10	SSR10: RDRF SSR10: ORE SSR10: FRE SSR10: PE	Reception interrupt request from UART/ CSIO/I^{2}C ch.10
0	1	1	Reception interrupt request from UART/CSIO/I^{2}C ch.11	SSR11: RDRF SSR11: ORE SSR11: FRE SSR11: PE	Reception interrupt request from UART/ CSIO/I^{2}C ch.11
1	0	0	Transmission interrupt request from UART/CSIO/I^{2}C ch.8 Transmission bus idle interrupt request Transmission FIFO interrupt request	SSR8: TDRE SSR8: TBI FCR18: FDRQ	-
1	0	1	Transmission interrupt request from UART/CSIO/I^{2}C ch.9 Transmission bus idle interrupt request Transmission FIFO interrupt request	SSR9: TDRE SSR9: TBI FCR19: FDRQ	-

MFSSEL2	MFSSEL1	MFSSEL0	Interrupt Request To Be Cleared by the DMAC		Channel To Be Used To Output a DMA Transfer Stop Request
			Interrupt Request Name	Flag Bit To Be Cleared	
1	1	0	Transmission interrupt request from UART/CSIO/I^{2}C ch.10 Transmission bus idle interrupt request Transmission FIFO interrupt request	SSR10: TDRE SSR10: TBI FCR110: FDRQ	-
1	1	1	Transmission interrupt request from UART/CSIO/I^{2}C ch.11 Transmission bus idle interrupt request Transmission FIFO interrupt request	SSR11: TDRE SSR11: TBI FCR111: FDRQ	-

* A reception interrupt request includes an interrupt request due to an error upon reception.

34.3.8 Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6)

The following interrupt requests are assigned to interrupt vector number 41 (decimal).

- Main timer interrupt request
- Sub timer interrupt request
- Underflow interrupt request from the watch counter

This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-8 shows the bit configuration of select register 6 for DMA transfer request clear by a peripheral function (ICSEL6).

Figure 34.3-8 The Bit Configuration of Select Register 6 for DMA Transfer Request Clear by a Peripheral Function (ICSEL6)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	MWSEL1	MWSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit1, bit0]: MWSEL1, MWSEL0 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 41 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 41 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

MWSEL1	MWSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Main timer interrupt request	MTMCR: MTIF
0	1	Sub timer interrupt request	STMCR: STIF
1	0	Underflow interrupt request from the watch counter	WCCR: WCIF
1	1		

34.3.9 Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7)

A priority conversion interrupt request and scan conversion interrupt request of the 10-bit A/D converter are assigned to interrupt vector number 42 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-9 shows the bit configuration of select register 7 for DMA transfer request clear by a peripheral function (ICSEL7).

Figure 34.3-9 Bit Configuration of the Select Register 7 for DMA Transfer Request Clear by a Peripheral Function (ICSEL7)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ADCSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit0]: ADCSEL0 (interrupt request select bit)

This bit selects the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 42 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 42 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Priority conversion interrupt request from unit 0 of the 10-bit A/D converter	ADCR0: SCIF
1	Scan conversion interrupt request from unit 0 of the 10-bit A/D converter	ADCR0: PCIF

34.3.10 Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8)

The interrupt request of the 32-bit input capture ch.0 to ch.3 is assigned to interrupt vector number 44 (decimal).
This register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-10 shows the bit configuration of select register 8 for DMA transfer request clear by a peripheral function (ICSEL8).

Figure 34.3-10 The Bit Configuration of Select Register 8 for DMA Transfer Request Clear by a Peripheral Function (ICSEL8)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	ICUSEL1	ICUSEL0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit1, bit0]: ICUSEL1, ICUSEL0 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 44 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 44 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

ICUSEL1	ICUSEL0	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Edge detection interrupt request from 32-bit input capture ch.0	ICS01: ICP0
0	1	Edge detection interrupt request from 32-bit input capture ch.1	ICS01: ICP1
1	0	Edge detection interrupt request from 32-bit input capture ch.2	ICS23: ICP2
1	1	Edge detection interrupt request from 32-bit input capture ch.3	ICS23: ICP3

34.3.11 Select Register 9 for DMA Transfer Request Clear by a Peripheral Function (ICSEL9)

The following interrupt requests are assigned to interrupt vector number 37 (decimal).

- 32-bit input capture ch.4 to ch.7
- Reception interrupt request from UART/CSIO/I^2C ch.7

From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-11 shows the bit configuration of select register 9 for DMA transfer request clear by a peripheral function (ICSEL9).

Figure 34.3-11 Bit Configuration of Select Register 9 for DMA Transfer Request Clear by a Peripheral Function (ICSEL9)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	ICUSEL12	ICUSEL11	ICUSEL10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2 to bit0]: ICUSEL12 to ICUSEL10 (interrupt request select bit)

These bits select the interrupt request to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 37 (decimal).

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 37 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

ICUSEL12	ICUSEL11	ICUSEL10	Explanation	
			Interrupt Request Name	Flag Bit To Be Cleared
0	0	0	Edge detection interrupt request from 32-bit input capture ch.4	ICS45: ICP4
0	0	1	Edge detection interrupt request from 32-bit input capture ch.5	ICS45: ICP5
0	1	0	Edge detection interrupt request from 32-bit input capture ch.6	ICS67: ICP6
0	1	1	Edge detection interrupt request from 32-bit input capture ch.7	ICS67: ICP7
1	0	0	Reception interrupt request from UART/CSIO/I^2C ch.7*	SSR7: RDRF SSR7: ORE SSR7: FRE SSR7: PE
1	0	1		
1	1	0		
1	1	1		

* A reception interrupt request includes an interrupt request due to an error upon reception.

34.3.12 Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10)

The interrupt request of the base timer ch.0 to ch.3 is assigned to interrupt vector number 46 to 49 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-12 shows the bit configuration of select register 10 for DMA transfer request clear by a peripheral function (ICSEL10).

Figure 34.3-12 The Bit Configuration of Select Register 10 for DMA Transfer Request Clear by a Peripheral Function (ICSEL10)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	BTSEL03	BTSEL02	BTSEL01	BTSEL00
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit4]: Reserved bits

In case of writing	Always write "0" to this bit (these bits).
In case of reading	A value is undefined.

[bit3]: BTSEL03 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 49 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 49 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.3	BT3STC: UDIR BT3STC: DTIR BT3STC: OVIR
1	Interrupt request 1 from base timer ch.3	BT3STC: TGIR BT3STC: EDIR

[bit2]: BTSEL02 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 48 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 48 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.2	BT2STC: UDIR BT2STC: DTIR BT2STC: OVIR
1	Interrupt request 1 from base timer ch.2	BT2STC: TGIR BT2STC: EDIR

[bit1]: BTSEL01 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 47 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 47 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.1	BT1STC: UDIR BT1STC: DTIR BT1STC: OVIR
1	Interrupt request 1 from base timer ch.1	BT1STC: TGIR BT1STC: EDIR

[bit0]: BTSEL00 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 46 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 46 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.0	BT0STC: UDIR BT0STC: DTIR BT0STC: OVIR
1	Interrupt request 1 from base timer ch.0	BT0STC: TGIR BT0STC: EDIR

34.3.13 Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11)

The interrupt requests from base timer ch.4 and ch.5 are assigned to interrupt vector number 50 (decimal).
The interrupt requests from base timer ch.6 and ch.7 are assigned to interrupt vector number 51 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-13 shows the bit configuration of select register 11 for DMA transfer request clear by a peripheral function (ICSEL11).

Figure 34.3-13 The Bit Configuration of Select Register 11 for DMA Transfer Request Clear by a Peripheral Function (ICSEL11)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	BTSEL13	BTSEL12	BTSEL11	BTSEL10
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit3, bit2]: BTSEL13, BTSEL12 (interrupt request select bit)

These bits select either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 51 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 51 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL13	BTSEL12	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.6	BT6STC: UDIR BT6STC: DTIR BT6STC: OVIR
0	1	Interrupt request 1 from base timer ch.6	BT6STC: TGIR BT6STC: EDIR
1	0	Interrupt request 0 from base timer ch.7	BT7STC: UDIR BT7STC: DTIR BT7STC: OVIR
1	1	Interrupt request 1 from base timer ch.7	BT7STC: TGIR BT7STC: EDIR

[bit1, bit0]: BTSEL11, BTSEL10 (interrupt request select bit)

These bits select interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 50 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 50 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL11	BTSEL10	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.4	BT4STC: UDIR BT4STC: DTIR BT4STC: OVIR
0	1	Interrupt request 1 from base timer ch.4	BT4STC: TGIR BT4STC: EDIR
1	0	Interrupt request 0 from base timer ch.5	BT5STC: UDIR BT5STC: DTIR BT5STC: OVIR
1	1	Interrupt request 1 from base timer ch.5	BT5STC: TGIR BT5STC: EDIR

34.3.14 Select Register 12 for DMA Transfer Request Clear by a Peripheral Function (ICSEL12)

The interrupt requests from base timer ch.8 and ch.9 are assigned to interrupt vector number 52 (decimal).
The interrupt requests from base timer ch.10 and ch.11 are assigned to interrupt vector number 53 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-14 shows the bit configuration of select register 12 for DMA transfer request clear by a peripheral function (ICSEL12).

Figure 34.3-14 Bit Configuration of the Select Register 12 for DMA Transfer Request Clear by a Peripheral Function (ICSEL12)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	BTSEL23	BTSEL22	BTSEL21	BTSEL20
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit4]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit3, bit2]: BTSEL23, BTSEL22 (interrupt request select bit)

These bits select either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 53 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 53 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL23	BTSEL22	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.10	BTASTC: UDIR BTASTC: DTIR BTASTC: OVIR
0	1	Interrupt request 1 from base timer ch.10	BTASTC: TGIR BTASTC: EDIR
1	0	Interrupt request 0 from base timer ch.11	BTBSTC: UDIR BTBSTC: DTIR BTBSTC: OVIR
1	1	Interrupt request 1 from base timer ch.11	BTBSTC: TGIR BTBSTC: EDIR

[bit1, bit0]: BTSEL21, BTSEL20 (interrupt request select bit)

These bits select either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 52 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 52 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL21	BTSEL20	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.8	BT8STC: UDIR BT8STC: DTIR BT8STC: OVIR
0	1	Interrupt request 1 from base timer ch.8	BT8STC: TGIR BT8STC: EDIR
1	0	Interrupt request 0 from base timer ch.9	BT9STC: UDIR BT9STC: DTIR BT9STC: OVIR
1	1	Interrupt request 1 from base timer ch.9	BT9STC: TGIR BT9STC: EDIR

34.3.15 Select Register 13 for DMA Transfer Request Clear by a Peripheral Function (ICSEL13)

The interrupt request from base timer ch.12 is assigned to interrupt vector number 54 (decimal). The interrupt request from base timer ch.13 is assigned to interrupt vector number 55 (decimal). From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-15 shows the bit configuration of select register 13 for DMA transfer request clear by a peripheral function (ICSEL13).

Figure 34.3-15 Bit Configuration of Select Register 13 for DMA Transfer Request Clear by a Peripheral Function (ICSEL13)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	BTSEL32	Reserved	BTSEL30
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit3]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit2]: BTSEL32 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 55 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 55 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.13	BTDSTC: UDIR BTDSTC: DTIR BTDSTC: OVIR
1	Interrupt request 1 from base timer ch.13	BTDSTC: TGIR BTDSTC: EDIR

[bit1]: Reserved bit

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit0]: BTSEL30 (interrupt request select bit)

This bit selects either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 54 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

Modes of Base Timer Operation	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 54 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

Written Value	Explanation	
	Interrupt Request Name	Flag Bit To Be Cleared
0	Interrupt request 0 from base timer ch.12	BTCSTC: UDIR BTCSTC: DTIR BTCSTC: OVIR
1	Interrupt request 1 from base timer ch.12	BTCSTC: TGIR BTCSTC: EDIR

34.3.16 Select Register 14 for DMA Transfer Request Clear by a Peripheral Function (ICSEL14)

The interrupt requests from base timer ch.14 and ch.15 are assigned to interrupt vector number 56 (decimal).
From the interrupt requests, this register selects an interrupt request that clears the flag bit with the DMA controller (DMAC).

Figure 34.3-16 shows the bit configuration of select register 14 for DMA transfer request clear by a peripheral function (ICSEL14).

Figure 34.3-16 Bit Configuration of Select Register 14 for DMA Transfer Request Clear by a Peripheral Function (ICSEL14)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	BTSEL41	BTSEL40
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

R/W: Read/Write

<Note>

Do not change this register during DMA transfer.

[bit7 to bit2]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	A value is undefined.

[bit1, bit0]: BTSEL41, BTSEL40 (interrupt request select bit)

These bits select either interrupt request 0 or 1 of the interrupt requests to clear the flag bit in the DMA controller (DMAC) from among the interrupt requests assigned to interrupt vector number 56 (decimal).

Each interrupt request assigned to Interrupt request 0 and Interrupt request 1 varies depending on the operation mode of the base timer.

	Interrupt Request 0	Interrupt Request 1
16/32-bit reload timer	Underflow interrupt request	Trigger interrupt request
16-bit PWM timer	Underflow interrupt request Duty match interrupt request	Trigger interrupt request
16/32-bit PWC timer	Overflow interrupt request	Measurement end interrupt request
16-bit PPG timer	Underflow interrupt request	Trigger interrupt request

When the DMA controller (DMAC) outputs an interrupt request clear signal to interrupt vector number 56 (decimal), the flag bit of the interrupt request selected with this bit will be cleared.

BTSEL41	BTSEL40	Explanation	
		Interrupt Request Name	Flag Bit To Be Cleared
0	0	Interrupt request 0 from base timer ch.14	BTESTC: UDIR BTESTC: DTIR BTESTC: OVIR
0	1	Interrupt request 1 from base timer ch.14	BTESTC: TGIR BTESTC: EDIR
1	0	Interrupt request 0 from base timer ch.15	BTFSTC: UDIR BTFSTC: DTIR BTFSTC: OVIR
1	1	Interrupt request 1 from base timer ch.15	BTFSTC: TGIR BTFSTC: EDIR

34.4 An Explanation of Operations and Setting Procedure Examples

This section explains operations and setting procedure examples for activating a DMA transfer by using an interrupt request of peripheral functions.

34.4.1 Operations upon a DMA Transfer

■ Setting

To select an interrupt request of a peripheral function as a transfer request source of DMA transfer, the interrupt vector number selections and the peripheral function settings are required.

The setting procedure is as follows:

1. Select the interrupt vector number (IO-data request registers [IORR0 to IORR7]).
   - Write values corresponding to the interrupt vector numbers to the bits ranging from IOS5 to IOS0.
   - Enable the activation of a DMA transfer in the IOE bit by using interrupt request from peripheral functions (IOE = 1).
2. Select an interrupt request to be cleared with the DMA controllers (DMAC) (select registers for DMA transfer request clear by a peripheral circuitry [ICSEL0 to ICSEL14]).
3. Set the DMA controller (DMAC).

   For details, see "CHAPTER 33 DMA Controller (DMAC)".
   - Set a transfer request source of DMA transfer to an interrupt request of the peripheral functions.
   - Enable DMA transfer operation and set it to a state of transfer request wait.
4. Set peripheral functions.

   See the chapters corresponding to the peripheral functions to be used.
   - Clear the flag of an interrupt request to be used for a DMA transfer.
   - Enable the generation of an interrupt request to be used for a DMA transfer.

<Notes>

- An interrupt request flag of peripheral functions is cleared by the DMA controller (DMAC). Therefore, it is not possible to use it as an interrupt request of peripheral functions.

  Set the interrupt level to "31" (interrupt is disabled) for an interrupt request to be used as a transfer request source of DMA transfer.

  For information on the interrupt level settings, see "CHAPTER 10 Interrupt Controller".
- When peripheral functions are set, clear an interrupt request flag first, and then enable the generation of an interrupt request.

■ Operation

Operations are as follows:

1. Peripheral functions are activated.
2. An interrupt request to be a DMA transfer request source is generated in the peripheral functions.
3. A DMA transfer request is generated, and the DMA controller (DMAC) is activated.
4. DMA transfer is finished.

   An interrupt request flag of the peripheral functions is cleared with the DMA controller (DMAC).

<Note>

Set the interrupt level so that the values of the interrupt level mask register (ILM) and interrupt control registers (ICR00 to ICR47) indicate the following values when interrupt requests are generated:

ILM ≤ **ICR**

If the value of the interrupt level mask registers (ILM) is greater than the value of the interrupt control registers (ICR00 to ICR47), the interrupt request generation operation of the peripheral functions will be established and also enable a DMA transfer request generation. However, this will make interrupt request processing operation unstable.

CHAPTER 35 Control of Built-in Program Memory

This series microcontrollers contain flash memory or mask ROM as built-in program memory.
This chapter explains the register settings for using the built-in program memory.

35.1 Overview of Built-in Program Memory Controller
35.2 Register for Built-in Program Memory Controller

35.1 Overview of Built-in Program Memory Controller

This series microcontrollers contain flash memory or mask ROM as built-in program memory.

■ Overview

The register shown below needs to be set to use the built-in program memory.

This register needs to be set regardless of whether the microcontroller uses flash memory products or mask ROM products.

- FLASH control register (FCTLR)

When flash memory products are used, see also "CHAPTER 36 Flash Memory".

■ Clock

Table 35.1-1 lists the clock used for the built-in program controller.

Table 35.1-1 Clock used for the built-in program controller

Clock Name	Description
Operation clock	Source clock (SRCCLK)

35.2 Register for Built-in Program Memory Controller

This section explains the register configuration and function of the built-in program memory controller.

■ Register for built-in program memory controller

Table 35.2-1 lists the register for the built-in program memory controller.

Table 35.2-1 Register for built-in program memory controller

Abbreviated Register Name	Register Name	Reference
FCTLR	FLASH control register	35.2.1

35.2.1 FLASH Control Register (FCTLR)

The FLASH control register controls access to the built-in program memory.

Figure 35.2-1 shows the bit configuration of the FLASH control register (FCTLR).

Figure 35.2-1 Bit Configuration of FLASH Control Register (FCTLR)

bit	15	14	13	12	11	10	9	8
	Reserved	FWE	Undefined	Undefined	FSZ1	FSZ0	FWC1	FWC0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	1	0	0	0	1	0	1	1

bit	7	0
	Reserved	
Attribute	R/W	
Initial value	0	

R/W: Read/Write

<Notes>

Built-in program memory cannot be accessed normally if a bit other than FWC1 or FWC0 of this register is rewritten during access. Be sure to rewrite the bit other than FWC1 or FWC0 of the register while the built-in program memory is not being accessed, as follows:

- At read access: Immediately before read operation
- At command issue: When the FRDY bit of the FLASH status register (FSTR) is "1"

Do not write an instruction to change the bit other than FWC1 or FWC0 of the register value, in a program running in the built-in program memory area. Rewrite the bit other than FWC1 or FWC0 of the register with a program running in a built-in RAM or an external area.

[bit15]: Reserved bit

In case of writing	Always write "1" to this (these) bit (bits).
In case of reading	"1" is read.

[bit14]: FWE(FLASH write enable)

- With a flash memory product
  This bit sets the access mode by enabling/disabling write to flash memory.

Written Value	Explanation
0	Disables write. CPU ROM mode is set.
1	Enables write. CPU programming mode is set.

- With a mask ROM product
  This bit is reserved bit.

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit13, bit12]: Undefined bits

In case of writing	Ignored
In case of reading	"0" is read.

[bit11, bit10]: FSZ1, FSZ0 (FLASH access size setting bits)

These bits set the size (bit width) of access to built-in program memory.

FS21	FS20	Access Size
0	0	Setting prohibited
0	1	16 bit
1	0	32 bit
1	1	Setting prohibited

Set FSZ [1:0] of FCTLR as "10" (32-bit) before use.

The access size that can be set varies depending on the access mode (read access or write access) of the built-in program memory.

	Read from	Written to
CPU ROM mode	32 bit	–
CPU programming mode	16 bit	16 bit

[bit9, bit8]: FWC1, FWC0(FLASH wait setting bits)

These bits set the interval (wait cycle) at which a request to read the built-in program memory is issued.

FWC1	FWC0	Wait Cycle
0	0	Setting prohibited
0	1	1
1	0	2
1	1	3

[bit7 to bit0]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

CHAPTER 36 Flash Memory

This chapter explains the functions and operations of flash memory.

36.1 Overview of Flash Memory
36.2 Flash Memory Configuration
36.3 Flash Memory Registers
36.4 Flash Memory Access Mode
36.5 Automatic Programming Algorithm
36.6 Explanation of Flash Memory Operation
36.7 Notes on Using Flash Memory

36.1 Overview of Flash Memory

The size of the flash memory built in this series microcontrollers is 512 KB.
Data can be erased in units of sectors or in all sectors altogether from the CPU. Data can also be written in units of half words.

■ Overview

The flash memory built in this series microcontrollers allows an access mode to be selected from 3 modes: 2 CPU modes and 1 ROM writer mode.

- CPU mode

  Flash memory is used as memory for storing CPU programs and data. The following two CPU modes are available:

  - CPU programming mode

    In this mode, flash memory data can be read, written, or erased (automatic programming algorithm[*1]). Because word access is disabled, programs in flash memory cannot be executed in this mode. Half word access is enabled.

  - CPU ROM mode

    In this mode, flash memory data is only read. Word access is enabled. This mode, however, does not support writing, erase, and activating the automatic programming algorithm.

- ROM writer mode

  The ROM writer can read, write, or erase flash memory data (automatic programming algorithm [*1])

*1 Automatic programming algorithm = Embedded Algorithm

<Note>

This manual describes flash memory that is used in CPU mode.

For details of access to flash memory from the ROM writer, see the instruction manual for the ROM writer used.

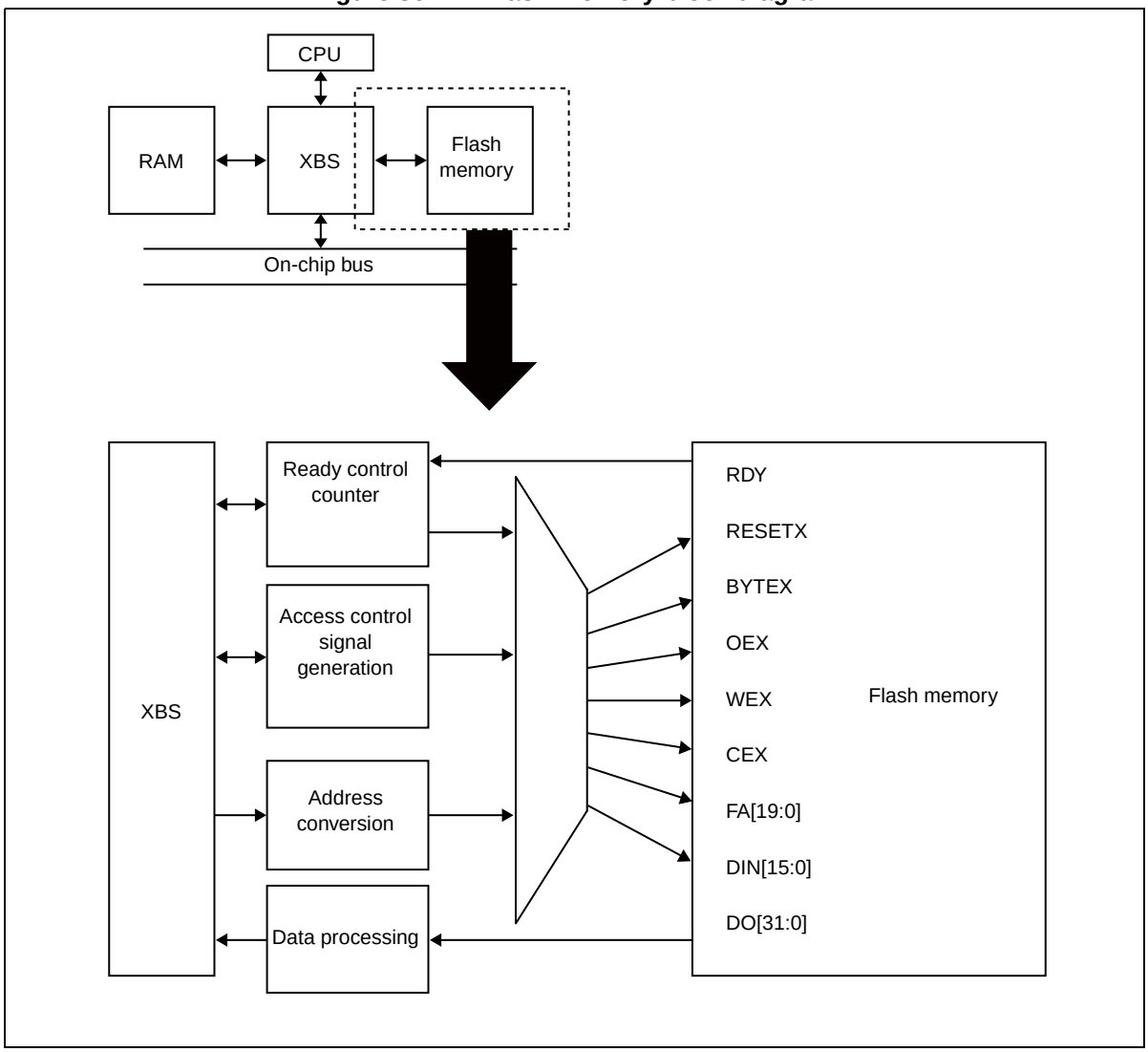
36.2 Flash Memory Configuration

This section explains the block configuration of flash memory.

■ Flash memory block diagram

Figure 36.2-1 is a flash memory block diagram.

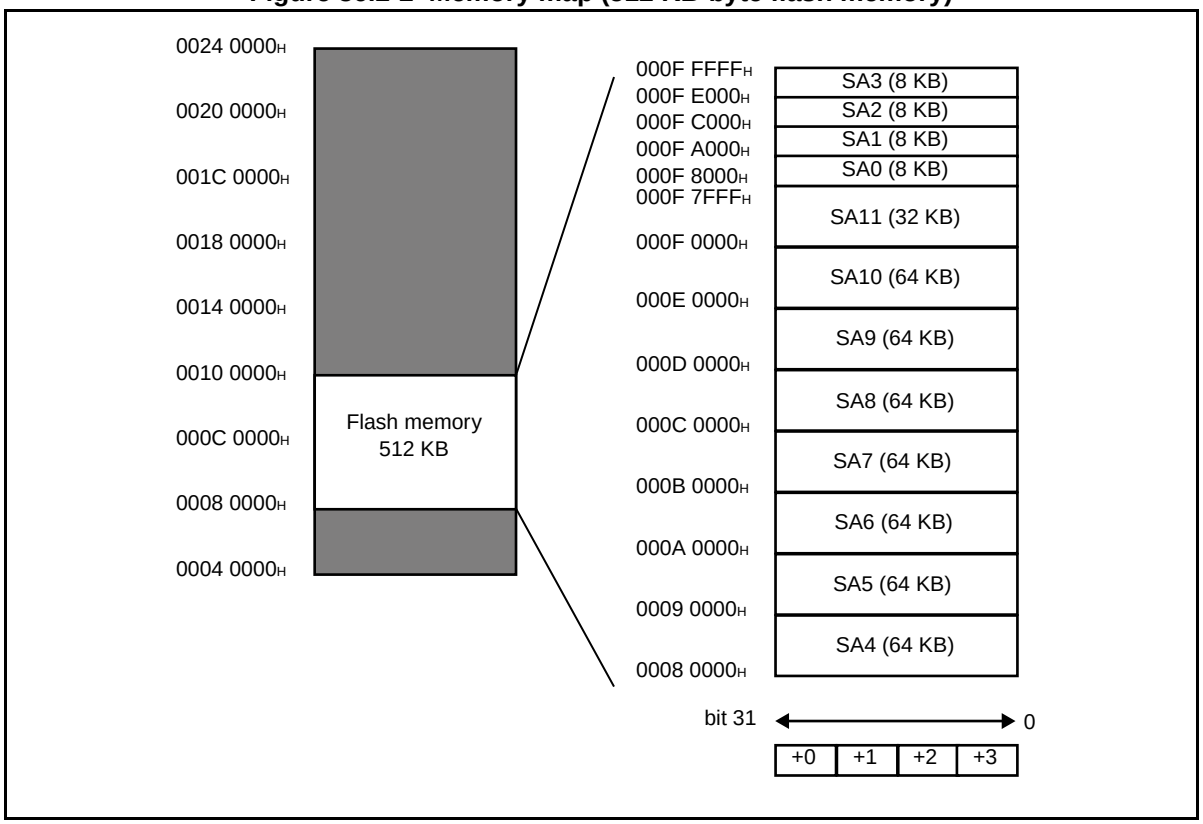
Figure 36.2-1 Flash memory block diagram



■ Flash memory sector configuration

The following shows the sector configuration of each capacity of flash memory.
Figure 36.2-2 shows the memory map of 512 KB flash memory.

Figure 36.2-2 Memory map (512 KB-byte flash memory)



■ Clock

Table 36.2-1 lists the clock used for flash memory.

Table 36.2-1 Clock used for flash memory

Clock Name	Description
Operation clock	Source clock (SRCCLK)

36.3 Flash Memory Registers

This section explains the configuration and functions of registers used for flash memory.

■ Flash memory registers

Table 36.3-1 lists the registers used for flash memory.

Table 36.3-1 Flash memory registers

Abbreviated Register Name	Register Name	Reference
FSTR	FLASH status register	36.3.1
FCTLR	FLASH control register	36.3.2

36.3.1 FLASH Status Register (FSTR)

This register indicates the state of flash memory.
Figure 36.3-1 shows the bit configuration of the FLASH status register (FSTR).

Figure 36.3-1 Bit configuration of FLASH status register (FSTR)

bit	7	6	5	4	3	2	1	0
	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FRDY
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

R: Read only

[bit7 to bit1]: Reserved bits

In case of writing	Always write "0" to this (these) bit (bits).
In case of reading	"0" is read.

[bit0]: FRDY (flash write enable bit)

This bit indicates whether flash memory writing or erase is in progress or complete using the automatic programming algorithm.When such operation is in progress, data cannot be written to or erased from flash memory.

Read Value	Explanation
0	In progress (writing/erase disabled)
1	Complete (writing/erase enabled)

36.3.2 FLASH Control Register (FCTLR)

The FLASH control register controls access to the built-in program memory.
For details of this register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".

36.4 Flash Memory Access Mode

The CPU accesses flash memory in one of the following two access modes:
- CPU programming mode
- CPU ROM mode

■ Overview

The access mode can be set using the FWE bit of the FLASH control register (FCTLR).

- CPU ROM mode (FWE=0)

  Flash memory data is only read in this mode. Because word access is enabled, 32-bit data can be read at one time.

  This mode, however, does not support writing or erasing commands or data, or activating the automatic programming algorithm.

<Note>

This mode is set upon the release of resetting.

- CPU programming mode (FWE=1)

  Data can be read, written, or erased in this mode. Because word access is disabled in this mode, programs in flash memory cannot run when this mode is enabled. The CPU performs access as shown below.

  - In case of reading

    The CPU performs half word accesses to flash memory to read 16 bits of data at one time.

  - In case of command writing

    The CPU activates the automatic programming algorithm to write or erase flash memory data.For details of the automatic programming algorithm, see "36.5 Automatic Programming Algorithm".

<Note>

CPU ROM mode is set when resetting is released during CPU operation. To enable this mode, write "1" to the FWE bit after releasing resetting. If resetting occurs after the CPU programming mode is set, the FWE bit changes to "0" and the ROM mode is restored.

36.5 Automatic Programming Algorithm

In CPU programming mode, the automatic programming algorithm is activated to write data to or erase data from flash memory.
This section explains the automatic programming algorithm.

36.5.1 Command Sequence

Writing half word (16-bit) data 1 to 6 times consecutively to flash memory activates the automatic programming algorithm. This operation is called the command. Table 36.5-1 lists the command sequence.

Table 36.5-1 Command Sequence

Command	Writing Count	1st Time		2nd Time		3rd Time		4th Time		5th Time		6th Time	
		Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
Read/Reset	1	$FXXXX_H$	$F0F0_H$	RA	RD	--	--	--	--	--	--	--	--
Read/Reset	4	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	$F0F0_H$	RA	RD	--	--	--	--
Writing	4	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	$A0A0_H$	PA	PD	--	--	--	--
Chip erase	6	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	8080_H	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	1010_H
Sector erase	6	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	$FAAAA_H$	8080_H	$FAAAA_H$	$AAAA_H$	$F5556_H$	5555_H	SA	3030_H
Sector erase suspended	1	$FXXXX_H$	$B0B0_H$	--	--	--	--	--	--	--	--	--	--
Sector erase restarting	1	$FXXXX_H$	3030_H	--	--	--	--	--	--	--	--	--	--
Continuous mode	3	$F5556_H$	$AAAA_H$	$FAAAA_H$	5555_H	$F5556_H$	2020_H	--	--	--	--	--	--
Continuous writing	2	$FXXXX_H$	$A0A0_H$	PA	PD	--	--	--	--	--	--	--	--
Reset continuous mode	2	$FXXXX_H$	9090_H	$FXXXX_H$	$F0F0_H$ or 0000_H	--	--	--	--	--	--	--	--

RA: Read address PA: Write address SA: Sector address (*) RD: Read data PA: Address PD: Write data

<Notes>

- Always write half word. (An address applicable in CPU mode is written.)
- If an invalid address or invalid data is written or if an address or data is written in the incorrect order, CPU programming mode is canceled and CPU ROM mode is restored.
- For details of the sector address, specify the low-order part of a 32-bit address space, the 4 lower bits of which indicate "2_H, "6_H ", "A_H", or "E_H ".

■ Read/ Reset Command

Sending the read/reset command listed in Table 36.5-1 to the target sector continuously can set the flash memory in the read/reset state.

Two types of read/reset commands, one including only 1 bus writing cycle and one including 4 bus writing cycles, are available. There are no essential differences between them.

When a read/reset command is issued, flash memory is kept in read state until another command is issued.

If a read/reset command is issued after execution of the automatic programming algorithm exceeds the timing limit, flash memory returns to the read/reset state. Read data from flash memory during a read cycle.

For details of the actual operation, see "36.6.1 Read/Reset Operation".

<Note>

When the device power is turned on, flash memory is automatically set in the read/reset state. In this case, a read/reset command need not be issued. Issue a read/reset command if another command fails to end normally or when the automatic programming algorithm needs to be initialized.

■ Program (Write) Command

Sending the write command listed in Table 36.5-1 four times consecutively to the target sector activates the automatic programming algorithm to write data to flash memory. Data can be written in any order of addresses or even beyond sector boundaries.

Write with half word in CPU programming mode.

After finishing writing four times as described in Table 36.5-1, the automatic programming algorithm is activated and then automatic writing to flash memory begins.

After execution of the automatic write algorithm command sequence, flash memory need not be controlled externally.

For details of the actual operation, see "36.6.2 Write Operation".

<Notes>

- If the fourth write command (write data cycle) is written at an odd-numbered location, data cannot be written normally. Always write it at an even-numbered location.
- 1 write command sequence can write only one piece of half word data. To write two or more data items, issue 1 write command sequence for each data item.

■ Chip Erase Command

Sending the chip erase command listed in Table 36.5-1 six times consecutively to the target sector can erase all sectors in flash memory all at once.

After writing six times as described in Table 36.5-1 is finished, the automatic programming algorithm is activated and then chip erase begins.

When the automatic erase algorithm is activated, the flash memory, before erasing the entire chip data, writes "0" to all cells in the chip to perform margin verification (preprogram). Therefore, the flash memory need not be written before chip erase.

During margin verification, the flash memory need not be controlled externally.

For details of the actual operation, see "36.6.3 Chip Erase".

■ Sector Erase Command

Sending the sector erase command listed in Table 36.5-1 six times consecutively to the target sector can erase the sector in flash memory.

When 50µs have passed after the sixth writing as listed in Table 36.5-1 is finished (timeout period), the automatic programming algorithm is activated and then sector erase begins.

To erase two or more sectors, write the erase code (3030_H) at the addresses of the sectors to be erased within 50µs (timeout period). If the next sector is not entered within the timeout period, the sector erase command may be disabled.

When the automatic erase algorithm is activated, the flash memory, before executing sector erase, writes "0" to the cells of the sector to be erased to perform margin verification (preprogram). Therefore, the flash memory need not be written before sector erase.

During margin verification, the flash memory need not be controlled externally.

For details of the actual operation, see "36.6.4 Sector Erase".

■ Sector Erase Suspend Command

If the sector erase suspend command listed in Table 36.5-1 is sent to the target sector during sector erase, sector erase is suspended so that data can be read from or written to sectors other than the one being erased.

If this command is issued during the timeout period after a sector erase command, the timeout is immediately finished and the erase operation is stopped. Note, however, that it takes a maximum of 20µs from when this command is issued to when sector erase is actually stopped.

For details of the actual operation, see "36.6.5 Sector Erase Suspending".

<Note>

This command is valid only during sector erase. The command is ignored if it is issued during chip erase or writing.

■ Sector Erase Restart Command

Sending the sector erase restart command listed in Table 36.5-1 consecutively to the target sector can release the sector erase suspended state and restart sector erase.

For details of the actual operation, see "36.6.6 Sector Erase Restarting".

<Note>

This command is valid only while sector erase is temporarily stopped with the sector erase suspend command. The command is ignored if it is issued during sector erase.

36.5.2 Execution State of Automatic Programming Algorithm

Flash memory implements writing and erase based on the automatic programming algorithm. The FRDY bit of the FLASH status register (FSTR) can be used to verify whether the automatic programming algorithm is in progress or the hardware sequence flag can be used to verify the operating state of the automatic programming algorithm.

For details of the FRDY bit of the FLASH status register (FSTR), see "36.3.1 FLASH Status Register (FSTR)".

■ Hardware Sequence Flag

The hardware sequence flag shows the state of the automatic programming algorithm. The operating status can be verified by reading an arbitrary address in flash memory when the FRDY bit of the FLASH status register (FSTR) is "0".

Figure 36.5-1 shows the bit configuration of the hardware sequence flag.

Figure 36.5-1 Bit configuration of hardware sequence flag

Half word access

bit	15	14	13	12	11	10	9	8
	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined

bit	7	6	5	4	3	2	1	0
	DPOLL	TOGG	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

Byte access

bit	7	6	5	4	3	2	1	0
	DPOLL	TOGG	TLOV	Undefined	SETI	TOGG2	Undefined	Undefined

<Notes>

- Word access is disabled for reading. Be sure to execute half word or byte access for reading in CPU programming mode.
- Even if an arbitrary address is read in CPU ROM mode, the hardware sequence flag cannot be read.
- Read an odd-numbered address for reading by byte access.

● Correspondence between bits and flash memory states

Table 36.5-2 lists the correspondence between hardware sequence flag bits and flash memory states.

Table 36.5-2 Correspondence between Flags and Flash Memory States

State			DPOLL	TOGG	TLOV	SETI	TOGG2
In progress	Writing		Inverted data *1	Toggle	0	0	1
	Sector/chip erase in progress		0	Toggle	0	1	Toggle
	Sector erase suspended	Read (Sector whose erase is suspended)	1	1	0	0	Toggle *2
		Read (Sector other than one whose erase is suspended)	Data *1	Data *1	Data *1	Data *1	Data *1
		Writing (Sector whose erase is suspended)	Inverted data *1	Toggle *3	1	0	1*4
Time limit exceeded	Sector/chip erase command		Inverted data *1	Toggle	1	0	1
	Sector erase suspended		0	Toggle	1	1	*5
	Write operation while sector erase is suspended		0	Toggle	1	1	*5

*1 For details of the values to be read, see "● Explanation of bits".

*2 Continuously reading addresses involved in the sector whose erase is suspended causes toggle operation that outputs "1" and "0" alternately.

*3 Regardless of the specified address, continuous reading causes toggle operation that outputs "1" and "0" alternately.

*4 Data may be written to a sector while erase of another sector is suspended. In this case, reading the sector being written outputs "1", or reading the sector whose erase is suspended causes toggle operation that outputs "1" and "0" alternately.

*5 Continuously reading a sector being written or erased while the TLOV bit is "1" causes toggle operation that outputs "1" and "0" alternately. However, reading another sector does not cause toggle operation.

● Explanation of bits

[bit15 to bit8]: Undefined bits

[bit7]: DPOLL: (data polling flag bit)

This bit indicates whether the automatic programming algorithm is in progress by the data polling function when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating status.

- In case of writing
  - During writing:

    The value (inverted data) opposite the value of bit7 of the data written last is read.

    The address specified for reading the hardware sequence flag is not accessed.
  - After writing is finished:

    The value of bit7 of the address specified for reading the hardware sequence flag is read.
- In case of sector erase
  - During sector erase: "0" is read from the sector being erased.
  - After sector erase: "1" is always read.
- In case of chip erase
  - During chip erase: "0" is always read.
  - After chip erase: "1" is always read.
- In case of section erase suspended

  The sector being suspended or erased can be verified by seeing this bit and TOGG bit.
  - When this bit is read by specifying the address of the sector being erased, in erase suspended read mode:

    "1" is read.
  - When this bit is read by specifying an address other than the address of the sector being erased, in erase suspended read mode:

    The value of bit7 of the specified address is read.
  - When this bit is read by specifying the address of the sector being erased, in erase suspended write mode:

    The value (inverted data) opposite the value of bit7 of the data in the sector being erased is read.

<Note>

While the automatic programming algorithm is activated, the data at the specified address cannot be read. Read the data after checking this bit to verify that the automatic programming algorithm has ended operation.

[bit6]: TOGG: (toggle flag bit)

This bit indicates whether the automatic programming algorithm is in progress when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating state.

- In case of writing, sector erase, or chip erase
  - During writing, sector erase, or chip erase

    If this bit is read continuously, "1" and "0" are read alternately (toggle operation).

    The address specified for reading the hardware sequence flag is not accessed.

- After writing, sector erase, or chip erase is finished:
  The value of bit6 of the address specified for reading the hardware sequence flag is read.

• In case of section erase suspended
  - When this bit is read by specifying the address of the sector being erased: "1" is read.
  - When this bit is read by specifying an address other than that of the sector being erased:
    The value of bit6 of the specified address is read.

<Notes>

- If an attempt is made to write to a write-protected sector, toggle operation is performed for about 2µs and then finished without rewriting data.
- If all of the selected sectors are write-protected during sector erase, toggle operation is performed for about 100µs and then the read/reset state is restored without erasing data.

[bit5]: TLOV: (timing limit overrun flag bit)

This bit indicates whether the execution time of the automatic programming algorithm exceeds the duration (internal pulse count) specified in the flash memory when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating state.

• In case of writing, sector erase, or chip erase
  One of the following values is read:

Read Value	Explanation
0	Within the time limit
1	Beyond the time limit

If the DPOLL or TOGG bit indicates that the automatic programming algorithm is in progress while this bit is "1", it indicates that writing or erase has failed.

For instance, because flash memory does not allow data "0" to be rewritten to "1", an attempt to write "1" to an address at which "0" has been written locks the flash memory, preventing the automatic programming algorithms from ending. In this event, the value of the DPOLL bit is kept invalid, and "1" and "0" are read alternately from the TOGG bit.

When the time limit is exceeded under this state, the TLOV bit changes to "1". If this bit becomes "1", issue a reset command.

<Note>

When this bit is "1", it indicates that flash memory has not been used correctly. It does not mean that the flash memory is faulty.

Issue a reset command and then take the appropriate action.

[bit4]: Undefined bit

[bit3]: SETI (sector erase timer flag bit)

When a sector is erased, a timeout period of 50µs is required from when the sector erase command is issued to when sector erase actually begins.

This bit indicates whether it is within the timeout period of the sector erase command when the hardware sequence flag is read with an arbitrary address specified.

The read value varies depending on the operating state.

- In case of sector erase:

  When sectors are erased, this bit can be checked before entering the next sector erase code to verify whether the next sector erase code can be accepted.

  The address specified for reading the hardware sequence flag is not accessed. Instead, the next value is read.

Read Value	Explanation
0	In a sector erase wait period The next sector erase code (3030_H) can be accepted.
1	The sector erase wait period is exceeded. *

* If the DPOLL or TOGG bit indicates that the automatic programming algorithm is in progress while this bit is "1", erase of flash memory internal data is started. In this case, the sector erase code (3030_H) and commands other than the erase suspend command are ignored until flash memory internal data is erased completely.

- In case of section erase suspended
  - When this bit is read by specifying the address of the sector being erased: "1" is read.
  - When this bit is read by specifying an address other than that of the sector being erased:

    The value of bit3 of the specified address is read.

[bit2]: TOGG2: (toggle flag bit)

This bit indicates whether erase is in progress or suspended when the hardware sequence flag is read with an arbitrary address specified. Because this bit indicates toggle operation in erase suspend read mode, verifying both this bit and TOGG bit can detect whether the read operation is performed (erase suspend read mode) while sector erase is suspended.

This bit can also be used to detect a sector in erase suspended state.

What this bit indicates differs depending on the operating state.

- In case of writing, sector erase, or chip erase (same as the TOGG bit)
  - During writing, sector erase, or chip erase

    If this bit is read continuously, "1" and "0" are read alternately (toggle operation).

    The address specified for reading the hardware sequence flag is not accessed.
  - After writing, sector erase, or chip erase is finished:

    The value of bit6 of the address specified for reading the hardware sequence flag is read.
- In case of section erase suspended:
  - When this bit is read by specifying an address of the sector in erase suspended state:

    If this bit is read continuously in erase suspend read mode, "1" and "0" are read alternately (toggle operation).
  - When this bit is read by specifying an address other than that of the sector in erase suspended state:

    In erase suspend write mode, "1" is read from this bit.

[bit1, bit0]: Undefined bits

36.6 Explanation of Flash Memory Operation

This section explains flash memory operations for each command.

■ Overview

Issuing a command 1 to 6 times consecutively for flash memory activates the automatic programming algorithm to perform the following operations:

- Read/Reset
- Writing
- Chip erase
- Sector erase
- Sector erase suspension
- Erase restart

The hardware sequence flag can be used to verify the execution state of the automatic programming algorithm.

For details of the commands and execution state of the automatic programming algorithm, see "36.5 Automatic Programming Algorithm".

36.6.1 Read/Reset Operation

This section explains the flash memory read/reset state.

Sending the read/reset command to the target sector continuously can set the flash memory in the read/reset state.

This state is the initial state of flash memory. Flash memory always returns to the read/reset state when the power is turned on or a command ends normally. At power on, a data read command need not be issued. Also, in the read/reset state, ordinary read access can be used to read data or implement program access from the CPU and, therefore, a read/reset command need not be issued to read data.

For details of the read/reset command, see "36.5 Automatic Programming Algorithm".

36.6.2 Write Operation

This section explains flash memory write operation.

■ Write Operation

The procedure for write operation is as follows:

1. Write commands are continuously sent to the target sector.

   The automatic programming algorithm is activated to write data to flash memory.

   After the write command is issued, flash memory need not be controlled externally.

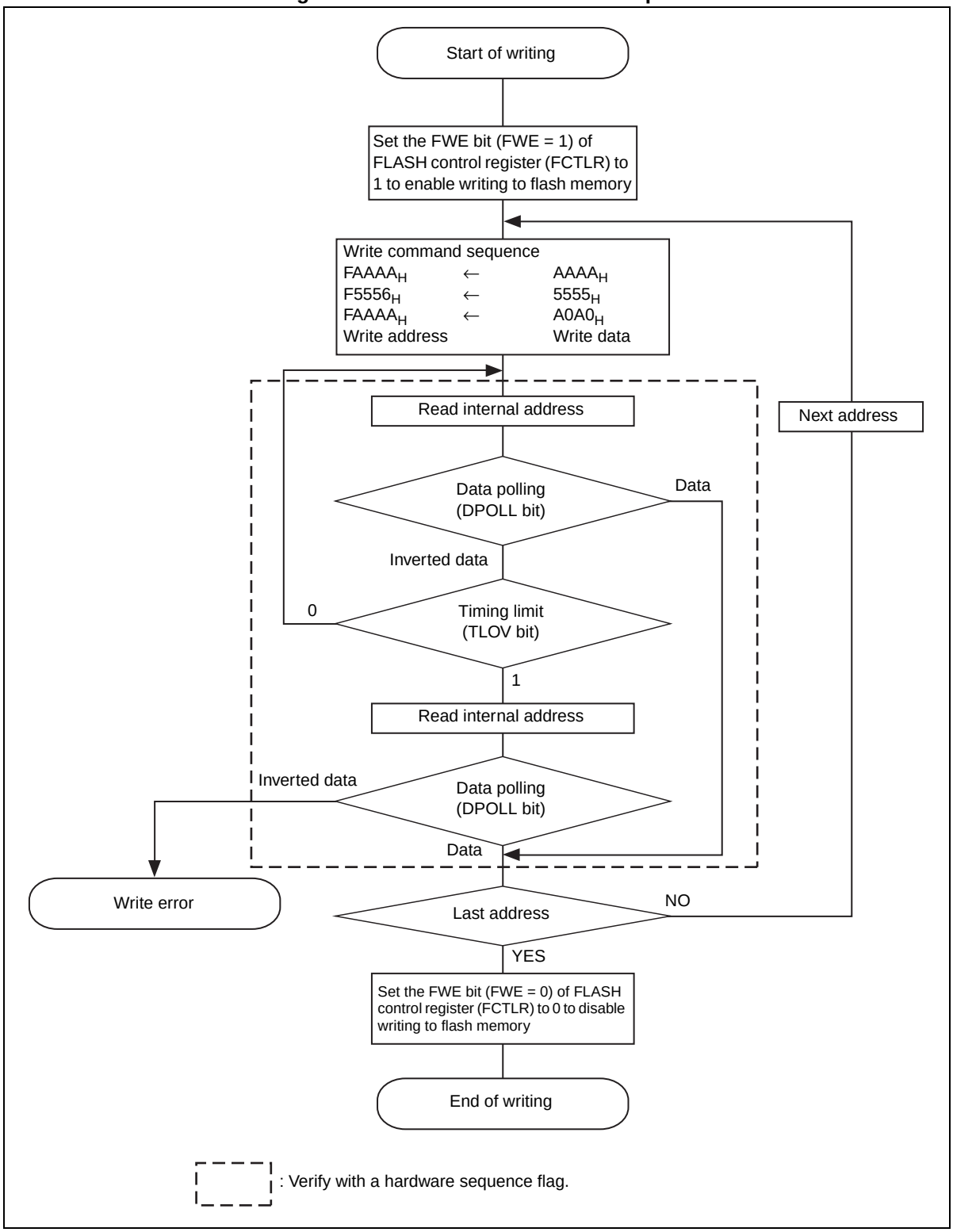
2. A read access is made to the address at which data was written.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data matches the written value, it means that writing to the flash memory is finished.

   If writing has not been finished, the value (inverted data) opposite the value of bit7 of the data written last is read.

Figure 36.6-1 shows an example of the flash memory write operation.

Figure 36.6-1 Write Procedure Example


Start of writing
Set the FWE bit (FWE = 1) of FLASH control register (FCTLR) to 1 to enable writing to flash memory
Write command sequence
FAAAA$_H$ ← AAAA$_H$
F5556$_H$ ← 5555$_H$
FAAAA$_H$ ← A0A0$_H$
Write address Write data
Read internal address
Next address
Data polling (DPOLL bit)
Data
Inverted data
0
Timing limit (TLOV bit)
1
Read internal address
Inverted data
Data polling (DPOLL bit)
Data
Write error
Last address
NO
YES
Set the FWE bit (FWE = 0) of FLASH control register (FCTLR) to 0 to disable writing to flash memory
End of writing
: Verify with a hardware sequence flag.

After writing is finished, flash memory returns to read mode and accepts no write address.

<Notes>

- For details of the write command, see "36.5 Automatic Programming Algorithm".
- Because the DPOLL bit of hardware sequence flag changes the value at the same time as the TLOV bit, it must be verified again even when the TLOV bit is "1".
- Toggle operation stops simultaneously when the TOGG and TLOV bit of the hardware sequence flag change to "1". Therefore, even when the TLOV bit is "1", the TOGG bit must be verified again.
- Data can be written to flash memory in any order of addresses or even beyond sector boundaries, but 1 write command sequence can write only 1 half word data item. To write two or more data items, issue 1 write command sequence for each data item.

■ Notes on Writing

- Data to which "0" is once written cannot be restored to "1". Rewriting "0" to "1" results in one of the following:
  - An element is judged as defective by the data polling algorithm.
  - The write time limit is exceeded and the TLOV bit of hardware sequence flag changes to "1".
  - It seems that "1" has been written.

  However, even when "1" seems to be written, the actual data remains "0" and therefore "0" is read by the read/reset mode command. To return data to "1", perform chip or sector erase.
- During write operation, the commands written to flash memory are all ignored.
- If the this device is reset during write operation, the data being written is not guaranteed.

36.6.3 Chip Erase

Flash memory sectors can be erased all at once. Erasing sectors all at once is called chip erase.

Sending chip erase commands continuously to target sectors can activate the automatic programming algorithm to erase all the sectors altogether.

For details of the chip erase command, see "36.5 Automatic Programming Algorithm".

1. Chip erase commands are continuously sent to the target sectors.

   The automatic programming algorithm is activated to write data to flash memory.

2. A read access is made to an arbitrary address.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", it means that chip erase is finished.

The duration required for chip erase is "sector erase time × total number of sectors + chip write time (preprogram)".

After chip erase is finished, flash memory returns to read/reset mode.

<Note>

When the automatic erase algorithm is activated, the flash memory, before erasing the entire chip data, writes "0" to all cells in the chip to perform margin verification (preprogram). Therefore, the flash memory need not be written before chip erase.

During margin verification, the flash memory need not be controlled externally.

36.6.4 Sector Erase

Only a specific sector that is selected from flash memory can be erased. Multiple sectors can be specified simultaneously.

The procedure for sector erase is as follows:

1. Sector erase commands are continuously sent to the target sectors.

   50 µs later (timeout period), the automatic programming algorithm is activated to start sector erase.

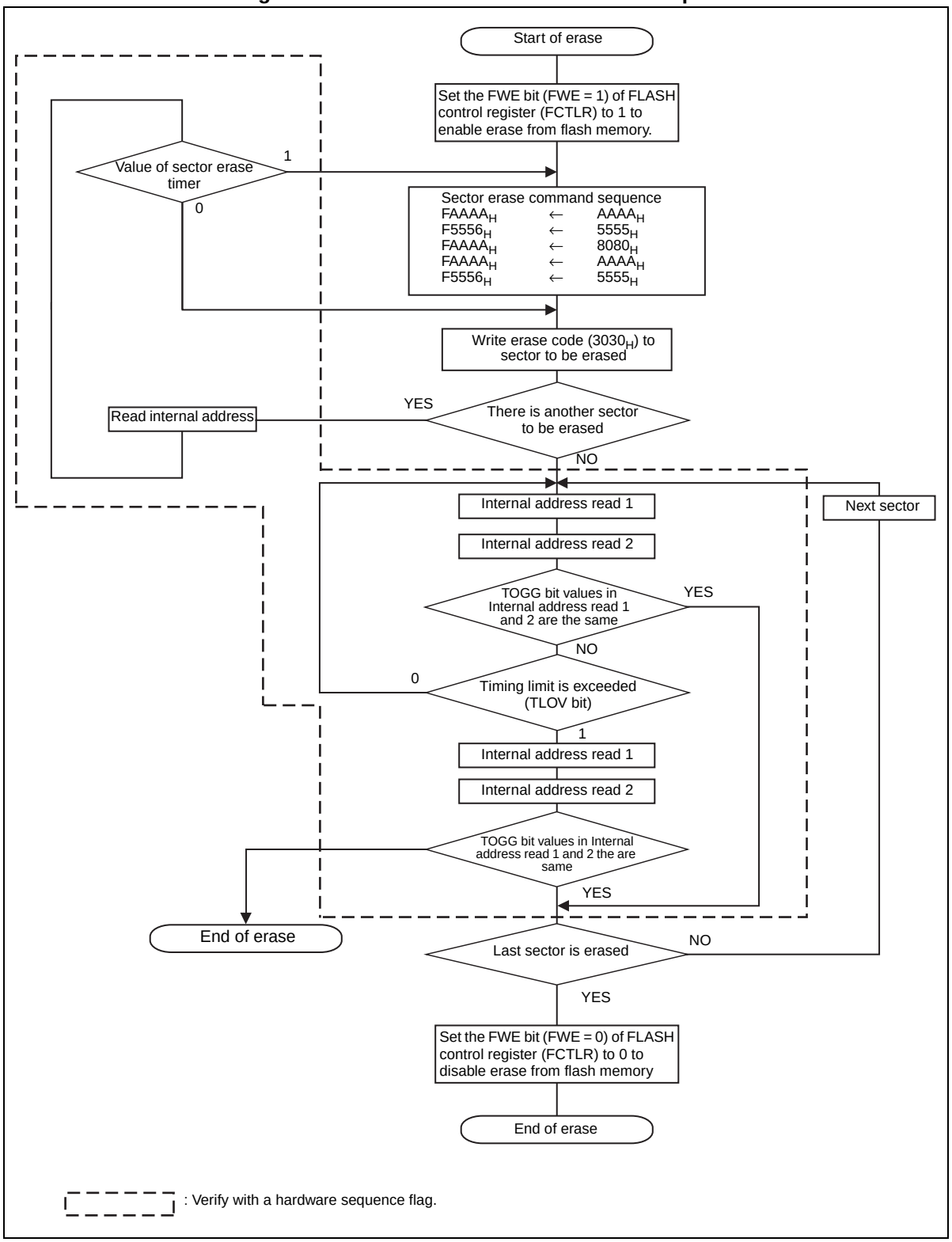
   To erase two or more sectors, write the erase code (3030_H) at the addresses of the sectors to be erased within 50µs (timeout period). If it is written after the lapse of the timeout period, the sector erase command may be invalid.

2. A read access is made to an arbitrary address.

   The read data becomes a hardware sequence flag. Therefore, when bit7 (DPOLL bit) of the read data is "1", it means that sector erase is finished.

The TOGG bit can also be used to verify whether sector erase is complete. Figure 36.6-2 shows an example of the sector erase procedure, in which the TOGG bit is used for verification.

Figure 36.6-2 Sector Erase Procedure Example


Start of erase
Set the FWE bit (FWE = 1) of FLASH control register (FCTLR) to 1 to enable erase from flash memory.
Value of sector erase timer
1
0
Sector erase command sequence
FAAAA$_H$ ← AAAA$_H$
F5556$_H$ ← 5555$_H$
FAAAA$_H$ ← 8080$_H$
FAAAA$_H$ ← AAAA$_H$
F5556$_H$ ← 5555$_H$
Write erase code (3030$_H$) to sector to be erased
YES
There is another sector to be erased
Read internal address
NO
Internal address read 1
Internal address read 2
Next sector
TOGG bit values in Internal address read 1 and 2 are the same
YES
NO
0
Timing limit is exceeded (TLOV bit)
1
Internal address read 1
Internal address read 2
TOGG bit values in Internal address read 1 and 2 the are same
YES
End of erase
Last sector is erased
NO
YES
Set the FWE bit (FWE = 0) of FLASH control register (FCTLR) to 0 to disable erase from flash memory
End of erase
: Verify with a hardware sequence flag.

The duration required for sector erase is "(sector erase time + sector write time (preprogram)) × number of sectors".

After sector erase is finished, flash memory returns to read/reset mode.

<Notes>

- For details of the sector erase command, see "36.5 Automatic Programming Algorithm".
- To specify a sector to be erased, specify an address (low-order side of 32 bits) whose 4 lower bits indicate 2_H, 6_H, A_H, or E_H.
- Because the DPOLL bit of hardware sequence flag changes the value at the same time as the TLOV bit, it must be verified again even when the TLOV bit is "1".
- The TOGG bit of the hardware sequence flag stops toggle operation simultaneously when the TLOV bit change to "1". Therefore, even when the TLOV bit is "1", the TOGG bit must be verified again.
- If a command other than the sector erase command and erase suspend command is issued during sector erase including the timeout period, flash memory is put in the read/reset state.

  In this event, flash memory is reset and, accordingly, the one or more sector erase commands that precede the issue of the command are disabled.

  To perform sector erase, issue sector erase commands again from the beginning.
- When the automatic erase algorithm is activated, the flash memory, before executing sector erase, writes "0" to the cells to be erased to perform margin verification (preprogram). Therefore, the flash memory need not be written before sector erase.

  During margin verification, the flash memory need not be controlled externally.

36.6.5 Sector Erase Suspending

Sector erase can be suspended so that data can be read from or written to sectors other than the sector being erased. Once sector erase is suspended, it is kept suspended until a sector erase restart command is issued.

If a sector erase suspend command is sent to the target sector while sector erase is stopped, sector erase is suspended so that data can be read from or written to sectors other than the sector being erased.

In this manual, reading data from another sector while sector erase is suspended is referred to as sector erase suspend read, and writing to another sector is referred to as sector erase suspend write.

■ Sector erase suspending

Sector erase is suspended by using the following procedure:

1. A sector erase suspend command is sent to the target sector during the period from the sector erase timeout period to sector erase.

   If the command is issued during the timeout period, the timeout period is immediately terminated and sector erase is stopped.

   If this command is issued during sector erase, it takes a maximum of 20 μs until sector erase is actually stopped.
2. A read access is made to the write address or the address at which sector erase was suspended.

   The read data becomes a hardware sequence flag. This means that sector erase is finished when "1" is read from bit7 (DPOLL bit) and bit6 (TOGG bit) of the read data.

   When sector erase is stopped, the FRDY bit of the FLASH status register (FSTR) changes to "1".

<Notes>

- For details of the sector erase suspend command, see "36.5 Automatic Programming Algorithm".
- Sector erase can be suspended only in the period from sector erase timeout period to sector erase in progress. Chip erase cannot be suspended. A sector erase suspend command is ignored if it is issued while sector erase is suspended.

■ State after sector erase is suspended

● Sector erase suspend read mode

After sector erase is suspended, sectors other than the one in the section erase suspended state can be read in the same way as usual. This state is referred to as sector erase suspend read mode.

<Notes>

The sector in the sector erase suspended state cannot be read. If an attempt is made to read the sector in the sector erase suspended state, a hardware sequence flag is read. When the hardware sequence flag is read, each bit of the read data is as follows:

- bit7 (DPOLL bit) and bit6 (TOGG bit): "1"
- bit2 (TOGG2 bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).

● Sector erase suspend write mode

If a program (write) command is issued in sector erase suspend read mode, data can be written to sectors other than the one in the sector erase suspend state. This state is referred to as sector erase suspend write mode.

Data can be written in the same way as in normal operation. Always write half word data.

<Notes>

- No data can be written to the sector in the sector erase suspend state.

  If an attempt is made to read the sector in the sector erase suspended state in sector erase suspend write mode, a hardware sequence flag is read. When the hardware sequence flag is read, each bit of the read data is as follows:

  - bit6 (TOGG bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
  - bit2 (TOGG2 bit): If this bit is read continuously, "1" and "0" are read alternately (toggle operation).
- If data is read from a sector other than the one whose erase is suspended in sector erase suspend write mode, bit 7 indicates the inverted value of the actual one.

36.6.6 Sector Erase Restarting

This section explains the operation for releasing the sector erase suspend state and restarting sector erase.

Sending a sector erase restart command to an arbitrary address while sector erase is suspended can restart sector erase.

When a sector erase restart command is issued, erase of the sector in the sector erase suspend state is restarted.

For details of the sector erase restart command, see "36.5 Automatic Programming Algorithm".

<Note>

The sector erase restart command is enabled only while sector erase is suspended. The sector erase restart command is ignored if it is issued during sector erase.

36.7 Notes on Using Flash Memory

Note the following points on using flash memory.

- If this device is reset during write operation, the data being written is not guaranteed.
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, do not run the program in the flash memory. The program fails to obtain normal values and hangs up.

  For details of the FLASH control register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".
- Do not generate an interrupt request when an interrupt vector table exists in the flash memory while the FWE bit of the FLASH control register (FCTLR) is set to 1. The program fails to obtain normal values and hangs up.

  For details of the FLASH control register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, do not change to Sub-run mode or a lowerpower dissipation mode.

  For details of the FLASH control register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".
- When the FWE bit of the FLASH control register (FCTLR) is set to 0 to enable CPU ROM mode (FWE = 0), do not write to flash memory.

  For details of the FLASH control register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".
- When the FWE bit of the FLASH control register (FCTLR) is set to 1 to enable CPU programming mode, be sure to write half words to flash memory. Do not write data in bytes.

  For details of the FLASH control register, see "35.2.1 FLASH Control Register (FCTLR)" in "CHAPTER 35 Control of Built-in Program Memory".
- Do not write to flash memory continuously. To continue to write to flash memory, be sure to insert at least 1 "NOP" instruction.
- After writing data to flash memory, be sure to read dummy data and then read data actually required. The data read immediately after it is written cannot be guaranteed.

CHAPTER 37 Wild Register

This chapter explains the functions and operations of the wild register.

37.1 Overview of Wild Register

37.2 Configuration of Wild Register

37.3 Registers of Wild Register

37.4 Explanation of Operations and Setting Procedure Examples of the Wild Register

37.5 Notes on Using the Wild Register

37.1 Overview of Wild Register

The wild register has the function to replace the data at a patch target address.
This series microcontroller has 16 built-in channels of wild register that enable 16 pairs of patch target addresses and replacement data to be set.

■ Overview

The wild register function can be used to read memory data (instruction code/data) at the specified address by replacing it with the data in the predetermined register.

With this function, the data to be read can be corrected without rewriting data in flash memory/ROM.

37.2 Configuration of Wild Register

This section explains the wild register configuration.

- Wild register address registers (WRAR00 to WRAR15)

  These registers are used to specify the addresses at which replacement data used by the wild register function is stored.
- Wild register data registers (WRDR00 to WRDR15)

  These registers are used to store replacement data.
- Wild register enable register (WREN)

  This register is used to enable/disable the wild register function.

37.3 Registers of Wild Register

This section explains the configuration and functions of registers used for the wild register.

■ Registers of wild register

Table 37.3-1 lists the registers used for the wild register.

Table 37.3-1 Registers of wild register (1 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
Common	WREN	Wild register enable register	37.3.3
0	WRAR00	Wild register address register 00	37.3.1
	WRDR00	Wild register data register 00	37.3.2
1	WRAR01	Wild register address register 01	37.3.1
	WRDR01	Wild register data register 01	37.3.2
2	WRAR02	Wild register address register 02	37.3.1
	WRDR02	Wild register data register 02	37.3.2
3	WRAR03	Wild register address register 03	37.3.1
	WRDR03	Wild register data register 03	37.3.2
4	WRAR04	Wild register address register 04	37.3.1
	WRDR04	Wild register data register 04	37.3.2
5	WRAR05	Wild register address register 05	37.3.1
	WRDR05	Wild register data register 05	37.3.2
6	WRAR06	Wild register address register 06	37.3.1
	WRDR06	Wild register data register 06	37.3.2
7	WRAR07	Wild register address register 07	37.3.1
	WRDR07	Wild register data register 07	37.3.2
8	WRAR08	Wild register address register 08	37.3.1
	WRDR08	Wild register data register 08	37.3.2
9	WRAR09	Wild register address register 09	37.3.1
	WRDR09	Wild register data register 09	37.3.2
10	WRAR10	Wild register address register 10	37.3.1
	WRDR10	Wild register data register 10	37.3.2

Table 37.3-1 Registers of wild register (2 / 2)

Channel	Abbreviated Register Name	Register Name	Reference
11	WRAR11	Wild register address register 11	37.3.1
	WRDR11	Wild register data register 11	37.3.2
12	WRAR12	Wild register address register 12	37.3.1
	WRDR12	Wild register data register 12	37.3.2
13	WRAR13	Wild register address register 13	37.3.1
	WRDR13	Wild register data register 13	37.3.2
14	WRAR14	Wild register address register 14	37.3.1
	WRDR14	Wild register data register 14	37.3.2
15	WRAR15	Wild register address register 15	37.3.1
	WRDR15	Wild register data register 15	37.3.2

37.3.1 Wild Register Address Register (WRAR00 to WRAR15)

These registers are used to specify the addresses at which replacement data used by the wild register function are stored. The value in WRAR21 to WRAR2 bits is compared with the actual address. When the memory at the address specified in this register is read, the actual memory is not read but instead the value set in the wild register data register (WRDR00 to WRDR15) is read.

Figure 37.3-1 shows the bit configuration of the wild register address register (WRAR00 to WRAR15).

Figure 37.3-1 Bit configuration of wild register address register (WRAR00 to WRAR15)

bit	31 to 22	21 to 2	1 to 0
	Undefined	WRAR21 to WRAR2	Undefined
Attribute	-	R/W	-
Initial value	X	X	X

R/W: Read/Write

-: Undefined

X: Undefined

<Notes>

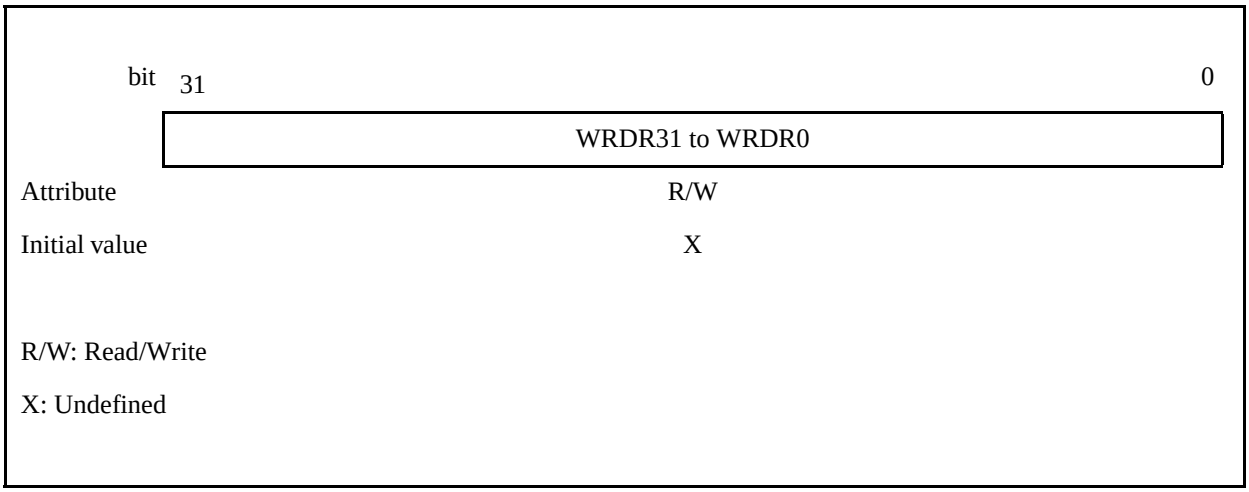
- Specify an address in units of words.
- This register cannot be read when the wild register function is enabled. If the register is read, the read value is undefined.
- Allocate the program, which sets an address in this register, to an area other than the built-in flash memory/ROM area.
- Be careful not to overlap the addresses to be set. The value read from an overlapped address is undefined.

37.3.2 Wild Register Data Register (WRDR00 to WRDR15)

These registers are used to store replacement data. When the memory at the address specified in the wild register address register (WRAR00 to WRAR15) is read, the actual memory is not read but instead the value set in this register is read.

Figure 37.3-2 shows the bit configuration of the wild register data register (WRDR00 to WRDR15).

Figure 37.3-2 Bit Configuration of Wild Register Data Register (WRDR00 to WRDR15)



<Notes>

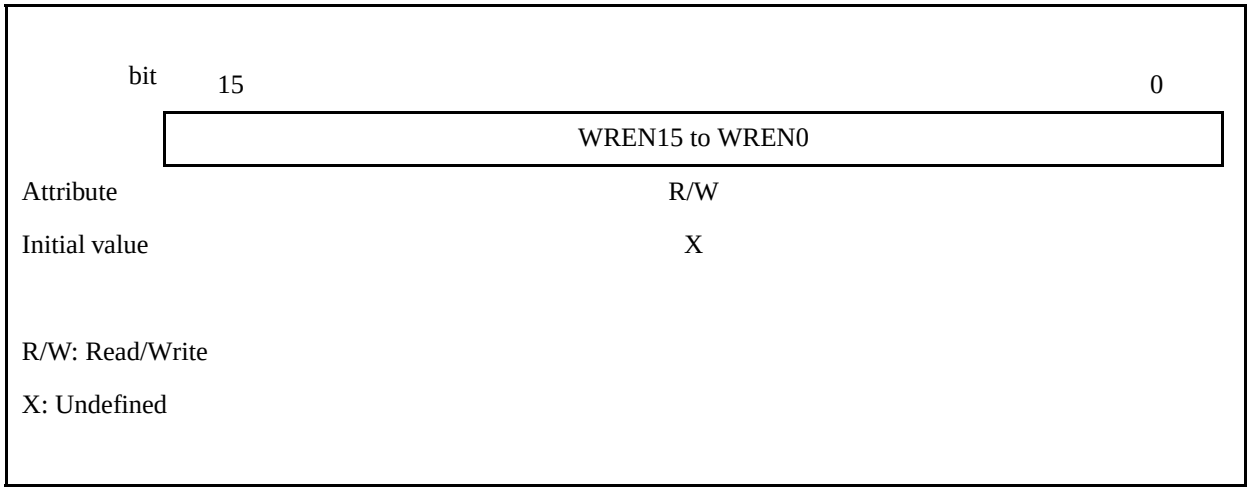
- Set word data in this register.
- This register cannot be read when the wild register function is enabled. If the register is read, the read value is undefined.

37.3.3 Wild Register Enable Register (WREN)

This register is used to enable/disable the wild register function.

Figure 37.3-3 shows the bit configuration of the wild register enable register (WREN).

Figure 37.3-3 Bit Configuration of Wild Register Enable Register (WREN)



[bit15 to bit0]: WREN15 to WREN0(operation enable bit)

Each bit enables or disable the wild register function of the corresponding channel.

The WREN15 bit corresponds to ch.15, the WREN14 bit corresponds to ch.14 ... the WREN0 bit corresponds to ch.0.

Written Value	Explanation
0	Disables the function.
1	Enables the function.

<Note>

Do not enable the wild register function during execution of the automatic algorithm of flash memory.

The FRDY bit of the FLASH status register (FSTR) can be used to verify that the automatic algorithm is operating (FRDY=0).

37.4 Explanation of Operations and Setting Procedure Examples of the Wild Register

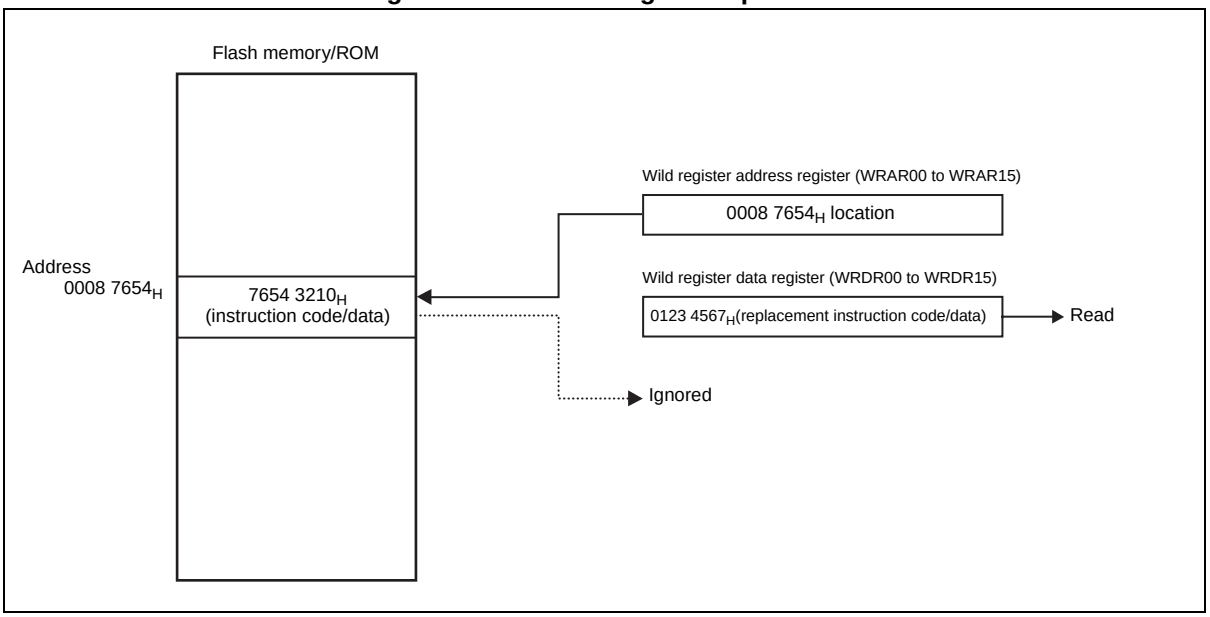
This section explains the operation of the wild register. The section also provides the setting procedure examples for operation.

37.4.1 Wild Register Operation

Figure 37.4-1 shows an example of wild register operation based on the following settings:

- Location 0008 7654_H is set in the wild register address register (WRAR00 to WRAR15).
- Wild register data register (WRDR00 to WRDR15) value: 0123 4567_H
- Value at location 0008 7654_H of flash memory/ROM: 7654 3210_H

Figure 37.4-1 Wild Register Operation



When the CPU attempts to read the data stored at location 0008 7654_H of flash memory/ROM, value "0123 4567_H" set in the wild register data register (WRDR0 to WRDR15) is read instead of "7654 3210_H" at location 0008 7654_H.

37.5 Notes on Using the Wild Register

Note the following points about using the wild register.

■ Notes on Programming

- Allocate the program, which sets an address in the wild register address register (WRAR00 to WRAR15), to an area other than the built-in flash memory/ROM area.
- Be careful not to overlap the addresses to be set in the wild register address register (WRAR00 to WRAR15). The value read from an overlapped address is undefined.
- Do not enable the wild register function during execution of the automatic algorithm of flash memory. The FRDY bit of the FLASH status register (FSTR) can be used to verify that the automatic algorithm is operating (FRDY=0).

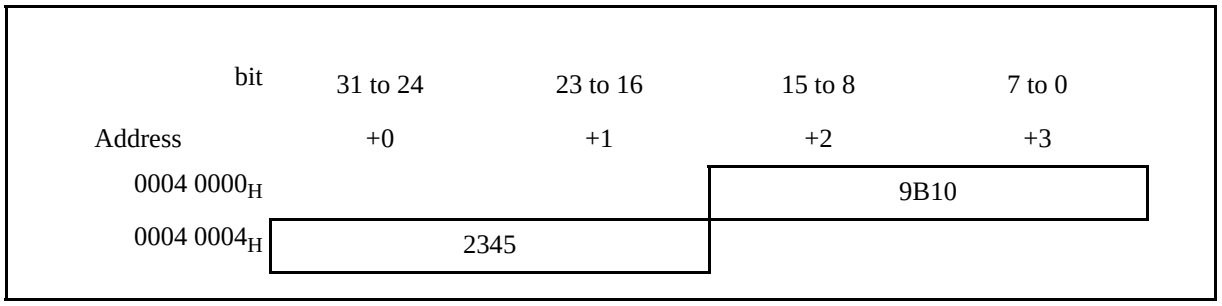
■ Notes on Operations

- The wild register address register (WRAR00 to WRAR15) and wild register data register (WRDR00 to WRDR15) are read in big endian mode.
- If a wild register is set at the address at which an instruction (32- or 48-bit instruction) exceeding 16 bits is stored, the CPU may not correctly interpret the instruction and malfunction. When setting a wild register at the address where a 32/48-bit instruction is allocated, do not set it inside the instruction.

  The following shows the operation of a 32/48-bit instruction.

  1. 32-bit instruction (LDI:20): Example) LDI:20 #0x12345,r0(9B102345$_H$)

Figure 37.5-1 Memory Map of the Register Allocated at Location 0004 0000$_H$



- When the wild register function is not used (WREN=0000)

  Data after replacement: 9B10 2345

  ❍ **0001 2345$_H$ is set in R0**.

- When 16 lower bits are replaced with the "INT" instruction
  (WRAR00=0004 0004, WRDR00=1FF4 ????, WREN=0001)

  Data after replacement: 9B10 **1FF4**

  ▼ **Because 1FF4$_H$ is interpreted not as an instruction but as LDI:20 immediate data, 0001 1FF4$_H$ is set in R0**

- When 16 upper bits are replaced with the INT instruction
  (WRAR00=0004 0000, WRDR00=1FF4???? , WREN=0001)

  Data after replacement: **1FF4** 2345

  ❍ **1FF4$_H$ is interpreted as an instruction.**

  ▼ **Next 2345$_H$ is interpreted not as LDI:20 immediate data, but as an instruction.**

2. 48-bit instruction (LDI:32) Example) LDI:32 #0x12345678,r0($9F8012345678_H$)

Figure 37.5-2 Memory Map of the Register Allocated at Location $0004\ 0000_H$/$0004\ 0004_H$

bit	31 to 24	23 to 16	15 to 8	7 to 0
Address	+0	+1	+2	+3
$0004\ 0000_H$	9F80		1234	
$0004\ 0004_H$	5678			

- When the wild register function is not used (WREN=0000)
  Data after replacement: 9F80 1234 5678
  ❍ **12345678_H is set in R0**.
- When 16 lower bits at location $0004\ 0000_H$ are replaced with the INT instruction (WRAR00=0004 0000, WRDR00=9F80 1FF4, WREN=0001)
  Data after replacement: 9F80 **1FF4** 5678
  ▼ **Because $1FF4_H$ is interpreted not as an instruction but as LDI:32 immediate data, 1FF4 5678_H is set in R0**
- When 16 upper bits at location $0004\ 0004_H$ are replaced with the INT instruction (WRAR00=0004 0004, WRDR00=1FF4 ????, WREN=0001)
  Data after replacement: 9F80 1234 **1FF4**
  ▼ **Because $1FF4_H$ is interpreted not as an instruction but as LDI:32 immediate data, 1234 $1FF4_H$ is set in R0**
- When 16 high-order bits at location $0004\ 0000_H$ are replaced with the INT instruction (WRAR00=0004 0004, WRDR00=1FF4 1234, WREN=0001)
  Data after replacement: **1FF4** 1234 5678
  ❍ **$1FF4_H$ is interpreted as an instruction.**
  ▼ **Next 1234_H and 5678_H are interpret not as LDI:32 immediate data, but as an instruction.**

CHAPTER 38 Serial Programming Connection

MB91F66x supports serial onboard write (Fujitsu Microelectronics standard) to flash memory.
This chapter explains the basic configuration for serial write to flash memory by using the Fujitsu Microelectronics Serial Programmer.

38.1 Fujitsu Microelectronics Serial Programmer

38.1 Fujitsu Microelectronics Serial Programmer

Fujitsu Microelectronics Serial Programmer (software) is an onboard programming tool for all Fujitsu Microelectronics-made controllers with built-in flash memory.
Two types of Serial Programmer are available according to the PC interface (RS-232C or USB) used. Choose the type according to your environment.
USB function is installed in the MB91F66x so that onboard write is possible by connecting the PC and microcontroller directly without performing USB-serial conversion.

■ Basic Configuration of FUJITSU MICROELECTRONICS MCU Programmer (Clock Asynchronous Serial Write)

FUJITSU MICROELECTRONICS MCU Programmer is used when the PC and microcontroller are connected through an RS-232C cable. MCU Programmer writes data, through clock asynchronous serial communication, to built-in flash memory of a microcontroller installed in the user system.

Figure 38.1-1 shows the basic configuration of FUJITSU MICROELECTRONICS MCU Programmer, and Table 38.1-1 lists the system configuration.

Figure 38.1-1 Basic Configuration of FUJITSU MICROELECTRONICS MCU Programmer

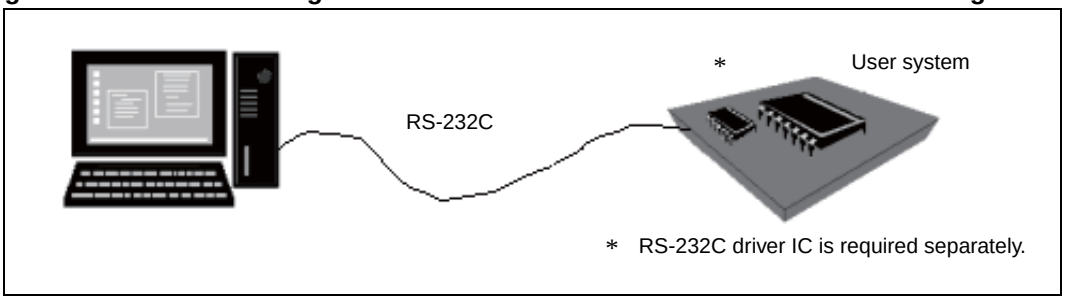


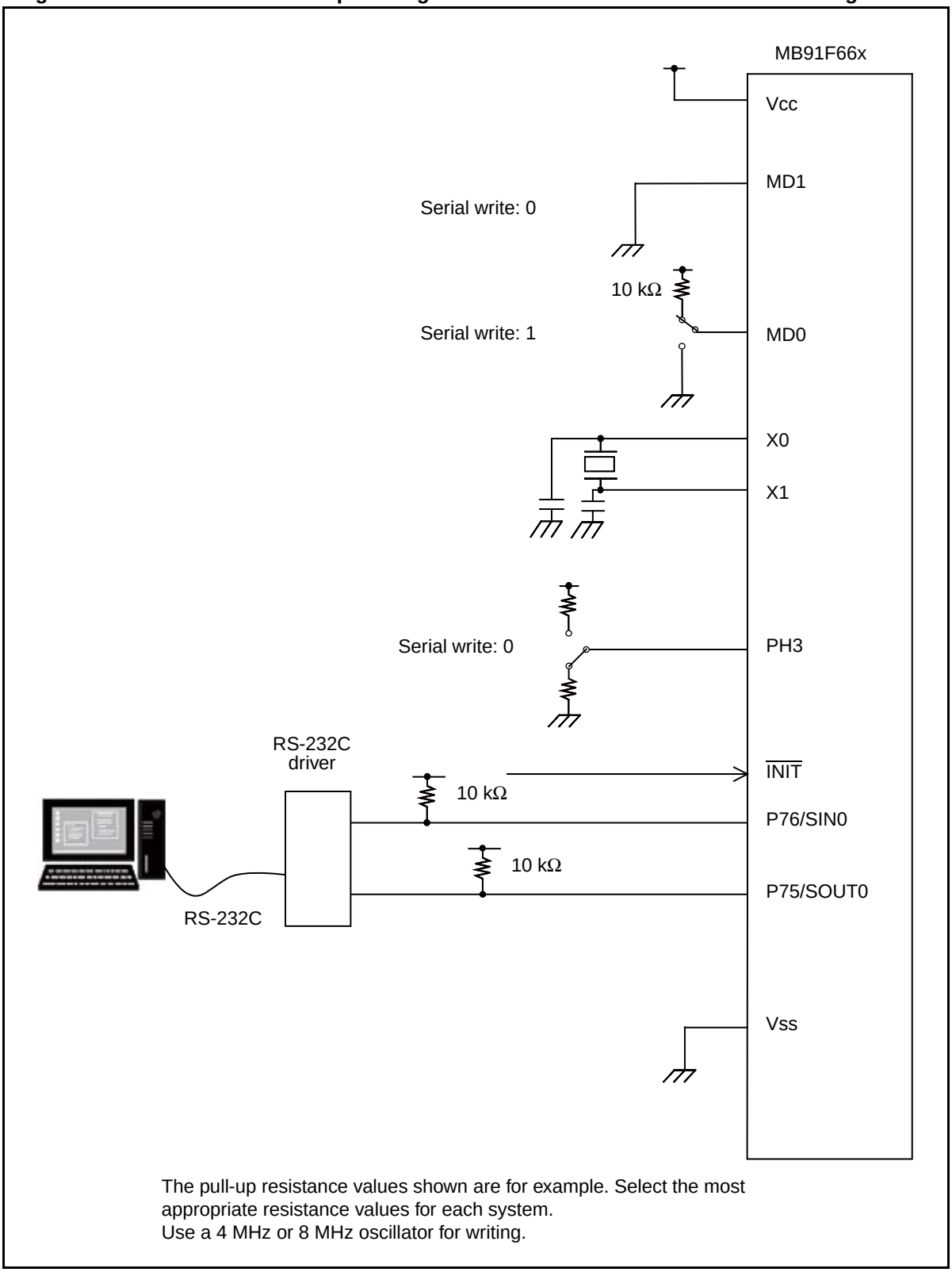
Table 38.1-1 System Configuration of FUJITSU MICROELECTRONICS MCU Programmer

Name	Type	Specifications
FUJITSU MICROELECTRONICS MCU Programmer	-	Software (can be downloaded from Web (registration system))*

* For registration, contact your sales representatives.

Figure 38.1-2 shows a connection example.

Figure 38.1-2 Connection Example using FUJITSU MICROELECTRONICS MCU Programmer



The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.
Use a 4 MHz or 8 MHz oscillator for writing.

Table 38.1-2 Oscillating frequency and communication baud rate available for clock asynchronous serial communication

Master Oscillating Frequency	Communication Baud Rate
4 MHz	9600 bps
8 MHz	19200 bps
16 MHz	38400 bps
24 MHz	57600 bps
48 MHz	115200 bps

■ Basic Configuration of FUJITSU MICROELECTRONICS USB Programmer (Clock Synchronous Serial Write)

FUJITSU MICROELECTRONICS USB Programmer is used when the PC and microcontroller are connected through an adapter (MB2146-09A-E). USB Programmer writes data, through clock synchronous serial communication, to built-in flash memory of a microcontroller.

Figure 38.1-3 shows the basic configuration of FUJITSU MICROELECTRONICS USB Programmer, and Table 38.1-3 lists the system configuration.

Figure 38.1-3 Basic Configuration of FUJITSU MICROELECTRONICS USB Programmer

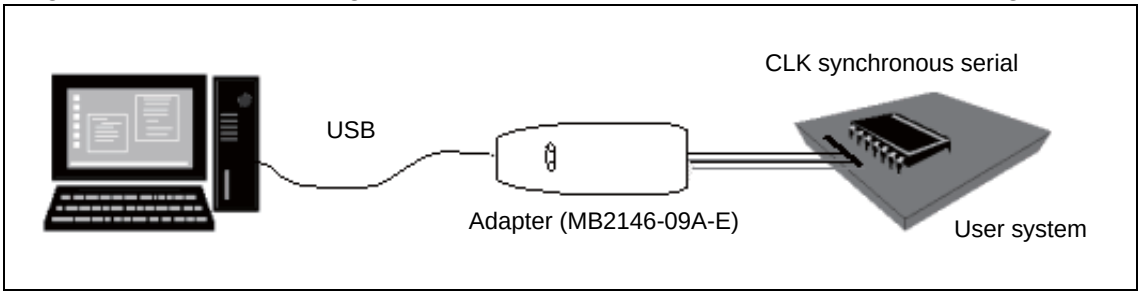


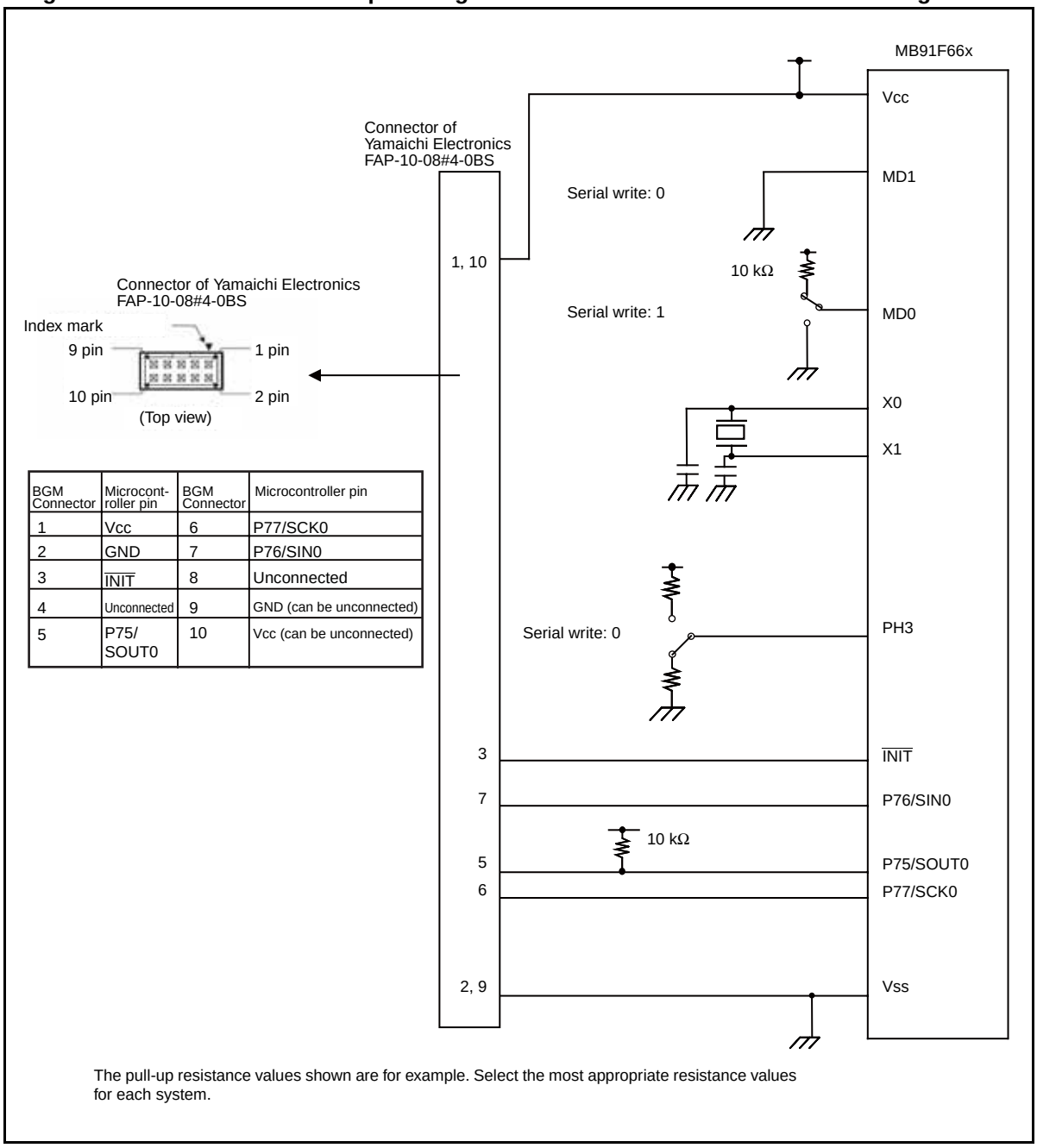
Table 38.1-3 System Configuration of FUJITSU MICROELECTRONICS USB Programmer

Name	Type	Specifications
FUJITSU MICROELECTRONICS USB Programmer	-	Software (can be downloaded from Web (registration system))*
Adapter	MB2146-09A-E	F^2MC family BGM adapter (Accessory: USB cable)

* For registration, contact your sales representatives.

Figure 38.1-4 shows a connection example.

Figure 38.1-4 Connection example using FUJITSU MICROELECTRONICS USB Programmer



BGM Connector	Microcont-roller pin	BGM Connector	Microcontroller pin
1	Vcc	6	P77/SCK0
2	GND	7	P76/SIN0
3	$\overline{\text{INIT}}$	8	Unconnected
4	Unconnected	9	GND (can be unconnected)
5	P75/ SOUT0	10	Vcc (can be unconnected)

The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.

■ Basic configuration of FUJITSU USB DIRECT Programmer (USB serial write)

FUJITSU USB DIRECT Programmer is used when the PC and microcontroller are connected through an USB cable. USB DIRECT Programmer writes data, through USB communication mode, to built-in flash memory of a microcontroller installed in the user system.

Figure 38.1-5 shows the basic configuration of FUJITSU USB DIRECT Programmer, and Table 38.1-4 lists the system configuration.

Figure 38.1-5 Basic configuration of FUJITSU USB DIRECT Programmer

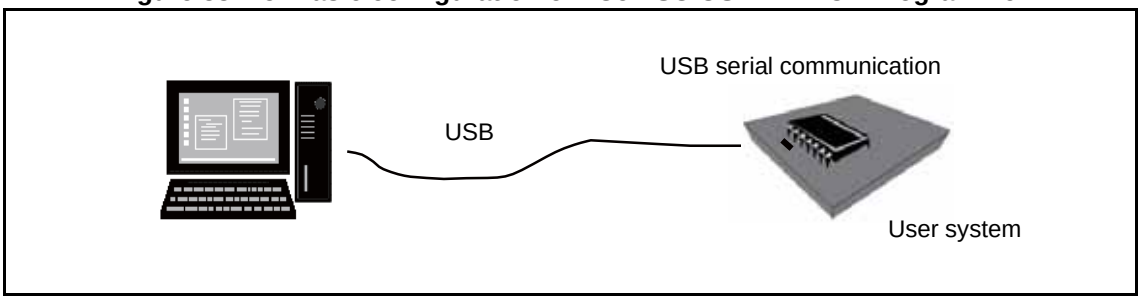


Table 38.1-4 System configuration of FUJITSU USB DIRECT Programmer

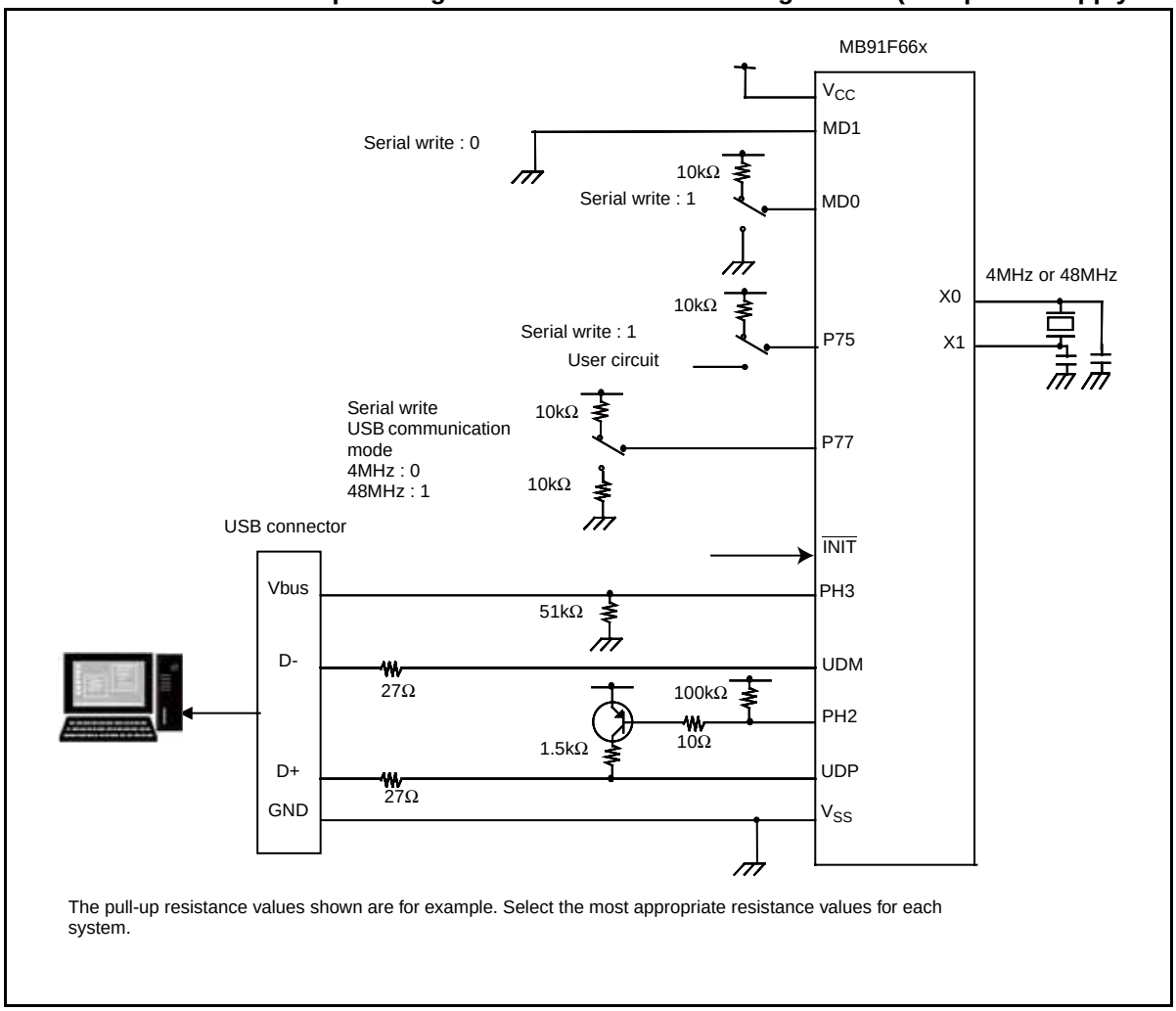
Name	Type	Specifications
FUJITSU USB DIRECT Programmer	-	Software (can be downloaded from Web (registration system))*
USB cable	-	Sold on the market

* For registration, contact your sales representatives.

For connection examples, see the manual (help section) of FUJITSU USB DIRECT Programmer.

Figure 38.1-6 shows a connection example.

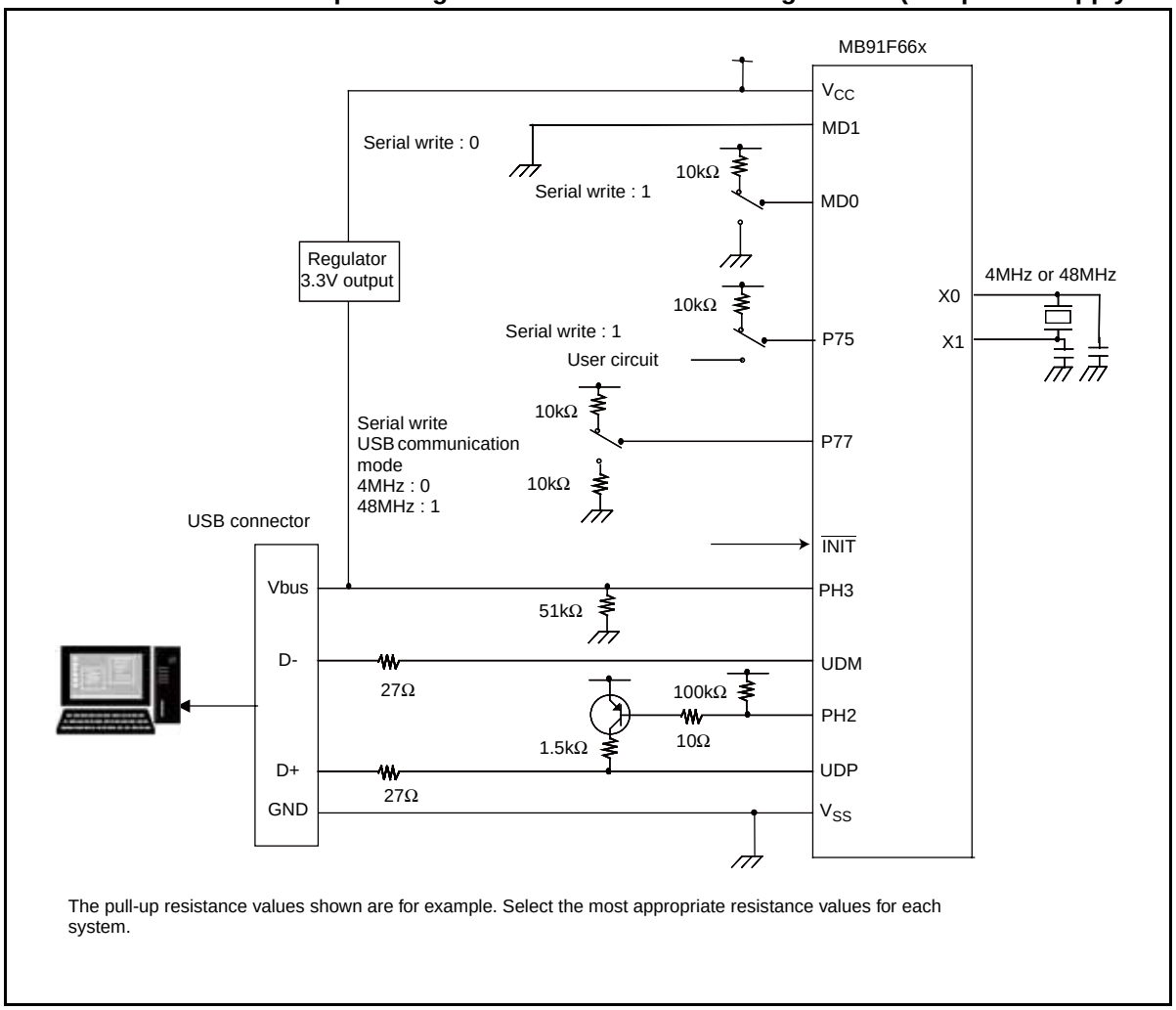
Figure 38.1-6 Connection example using FUJITSU USB DIRECT Programmer (own power supply is used)



The pull-up resistance values shown are for example. Select the most appropriate resistance values for each system.

Figure 38.1-7 shows a connection example.

Figure 38.1-7 Connection example using FUJITSU USB DIRECT Programmer (bus power supply is used)



38.1.1 Pins Used

Table 38.1-5 Pins used

Pins	Function	Supplement
MD1,MD0	Mode pin	Resetting the system ($\overline{INIT}$: L → H) with SOUT0=H after setting MD1=L and MD0=H enters the serial write mode. When attaching a pull-up or pull-down resistor, avoid wiring.
X0,X1	Oscillation pin	See the "Data Sheet" for the source clock frequencies that can be used in serial write mode. (Restrictions apply to clock asynchronous communication. For details, see Table 38.1-2.)
P75/SOUT0	Serial write mode activation pin/UART serial data output pin	After adding an external pull-up resistor and then releasing the reset state, setting the level of this pin to "H" activates the serial write mode. When the communication mode is set to UART, this pin becomes a serial data output pin when communication begins after the serial write mode is activated. When the communication mode is set to USB, the output operation can not be performed with keeping the pull-up state.
P76/SIN0	Clock synchronous/ asynchronous select pin/ UART serial data input pin	Setting the input level of this pin to "H" until the start of communication enables the clock asynchronous communication mode, and setting it to "L" enables the clock synchronous communication mode. This pin is used as an UART serial data input pin when communication begins after the serial write mode is activated.
P77/SCK0	Serial clock I/O pin	This pin becomes a serial clock input/output pin when the communication mode is set to clock synchronous communication. When the communication mode is USB, this pin controls the frequency for X0/X1. P77=L: X0/X1 = 4MHz P77=H: X0/X1 = 48MHz
PH3	Communication mode select pin	The communication mode is determined by the input level of this pin at reset to shift to the serial write mode. Setting this pin to "H" enables the USB communication mode, and setting it to "L" enables the UART communication mode.
PH2	Pull-up control pin for UDP	This pin controls the pull-up of USB side (+) when the communication mode is USB. PH2=L: Connect the pull-up resistor PH2=H: Disconnect the pull-up resistor
UDP	USB I/O pin	This pin becomes an input/output pin of USB side (+) when the communication mode is set to USB.
UDM	USB I/O pin	This pin becomes an input/output pin of USB side (-) when the communication mode is set to USB.

Table 38.1-5 Pins used

Pins	Function	Supplement
$\overline{INIT}$	Reset pin	-
V_{CC}	Supply voltage pin	For writing, supply power voltage to the microcontroller from the user system.
V_{SS}	GND pin	-

CHAPTER 39 Handling the Device

This chapter provides notes on using this series.

39.1 Notes on Handling the Device

39.1 Notes on Handling the Device

Note the following points on using this series.

■ PRECAUTIONS FOR HANDLING THE DEVICES

Any semiconductor devices have inherently a certain rate of failure. The possibility of failure is greatly affected by the conditions in which they are used (circuit conditions, environmental conditions, etc.). This page describes precautions that must be observed to minimize the chance of failure and to obtain higher reliability from your FUJITSU MICROELECTRONICS devices.

1. Precautions for Product Design

   This section describes precautions when designing electronic equipment using semiconductor devices.

- Absolute Maximum Ratings

  Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc.) in excess of absolute maximum ratings. Do not exceed these ratings.

- Recommended Operating Conditions

  The recommended operating conditions are required in order to ensure the normal operation of the semiconductor device. All of the device's electrical characteristics are warranted when the device is operated within these ranges.

  Always use semiconductor devices within their recommended operating condition ranges. Operation outside these ranges may adversely affect reliability and could result in device failure. No warranty is made with respect to uses, operating conditions, or combinations not represented on the data sheet. Users considering application outside the listed conditions are advised to contact their sales representatives beforehand.

- Processing and Protection of Pins

  These precautions must be followed when handling the pins which connect semiconductor devices to power supply and input/output functions.

  1. Preventing Over-Voltage and Over-Current Conditions

     Exposure to voltage or current levels in excess of maximum ratings at any pin is likely to cause deterioration within the device, and in extreme cases leads to permanent damage of the device. Try to prevent such overvoltage or over-current conditions at the design stage.

  2. Protection of Output Pins

     Shorting of output pins to supply pins or other output pins, or connection to large capacitance can cause large current flows. Such conditions if present for extended periods of time can damage the device. Therefore, avoid this type of connection.

  3. Handling of Unused Input Pins

     Unconnected input pins with very high impedance levels can adversely affect stability of operation. Such pins should be connected through an appropriate resistance to a power supply pin or ground pin.

- Latch-up

  Semiconductor devices are constructed by the formation of P-type and N-type areas on a substrate. When subjected to abnormally high voltages, internal parasitic PNPN junctions (called thyristor structures) may be formed, causing large current levels in excess of several hundred mA to flow continuously at the power supply pin. This condition is called latch-up.

  Note: The occurrence of latch-up not only causes loss of reliability in the semiconductor device, but can cause injury or damage from high heat, smoke or flame. To prevent this from happening, do the following:

(1) Be sure that voltages applied to pins do not exceed the absolute maximum ratings. This should include attention to abnormal noise, surge levels, etc.

(2) Be sure that abnormal current flows do not occur during the power-on sequence.

- Observance of Safety Regulations and Standards

  Most countries in the world have established standards and regulations regarding safety, protection from electromagnetic interference, etc. Customers are requested to observe applicable regulations and standards in the design of products.

- Fail-Safe Design

  Any semiconductor devices have inherently a certain rate of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.

- Precautions Related to Usage of Devices

  FUJITSU MICROELECTRONICS devices are intended for use in standard applications (computers, office automation and other office equipment, industrial, communications, and measurement equipment, personal or household devices, etc.).

  CAUTION: Customers considering the use of our products in special applications where failure or abnormal operation may directly affect human lives or cause physical injury or property damage, or where extremely high levels of reliability are demanded (such as aerospace systems, atomic energy controls, sea floor repeaters, vehicle operating controls, medical devices for life support, etc.) are requested to consult with sales representatives before such use. The company will not be responsible for damages arising from such use without prior approval.

2. Precautions for Package Mounting

   Package mounting may be either lead insertion type or surface mount type. In either case, for heat resistance during soldering, you should only mount under FUJITSU MICROELECTRONICS's recommended conditions. For detailed information about mount conditions, contact your sales representative.

- Lead Insertion Type

  Mounting of lead insertion type packages onto printed circuit boards may be done by two methods: direct soldering on the board, or mounting by using a socket.

  Direct mounting onto boards normally involves processes for inserting leads into through-holes on the board and using the flow soldering (wave soldering) method of applying liquid solder.

  In this case, the soldering process usually causes leads to be subjected to thermal stress in excess of the absolute ratings for storage temperature. Mounting processes should conform to FUJITSU MICROELECTRONICS recommended mounting conditions.

  If socket mounting is used, differences in surface treatment of the socket contacts and IC lead surfaces can lead to contact deterioration after long periods. For this reason it is recommended that the surface treatment of socket contacts and IC leads be verified before mounting.

- Surface Mount Type

  Surface mount packaging has longer and thinner leads than lead-insertion packaging, and therefore leads are more easily deformed or bent. The use of packages with higher pin counts and narrower pin pitch results in increased susceptibility to open connections caused by deformed pins, or shorting due to solder bridges.

  You must use appropriate mounting techniques. FUJITSU MICROELECTRONICS recommends the solder reflow method, and has established a ranking of mounting conditions for each product. Users are advised to mount packages in accordance with FUJITSU MICROELECTRONICS ranking of recommended conditions.

- Lead-Free Packaging

  Note: When ball grid array (BGA) packages with Sn-Ag-Cu balls are mounted using Sn-Pb eutectic soldering, junction strength may be reduced under some conditions of use.

- Storage of Semiconductor Devices

  Because plastic chip packages are formed from plastic resins, exposure to natural environmental conditions will cause absorption of moisture. During mounting, the application of heat to a package that has absorbed moisture can cause surfaces to peel, reducing moisture resistance and causing packages to crack. To prevent, do the following:

  1. Avoid exposure to rapid temperature changes, which cause moisture to condense inside the product. Store products in locations where temperature changes are slight.
  2. Use dry boxes for product storage. Products should be stored below 70% relative humidity, and at temperatures between 5°C and 30 °C. When you open Dry Package that recommends humidity 40% to 70% relative humidity.
  3. When necessary, FUJITSU MICROELECTRONICS packages semiconductor devices in highly moisture-resistant aluminum laminate bags, with a silica gel desiccant. Devices should be sealed in their aluminum laminate bags for storage.
  4. Avoid storing packages where they are exposed to corrosive gases or high levels of dust.

- Baking

  Packages that have absorbed moisture may be de-moisturized by baking (heat drying). Follow the FUJITSU MICROELECTRONICS recommended conditions for baking.

  Condition: 125 °C/24 h

- Static Electricity

  Because semiconductor devices are particularly susceptible to damage by static electricity, you must take the following precautions:

  1. Maintain relative humidity in the working environment between 40% and 70%. Use of an apparatus for ion generation may be needed to remove electricity.
  2. Electrically ground all conveyors, solder vessels, soldering irons and peripheral equipment.
  3. Eliminate static body electricity by the use of rings or bracelets connected to ground through high resistance (on the level of 1 MΩ). Wearing of conductive clothing and shoes, use of conductive floor mats and other measures to minimize shock loads is recommended.
  4. Ground all fixtures and instruments, or protect with anti-static measures.
  5. Avoid the use of styrofoam or other highly static-prone materials for storage of completed board assemblies.

3. Precautions for Use Environment

   Reliability of semiconductor devices depends on ambient temperature and other conditions as described above. For reliable performance, do the following:

   1. Humidity

      Prolonged use in high humidity can lead to leakage in devices as well as printed circuit boards. If high humidity levels are anticipated, consider anti-humidity processing.

   2. Discharge of Static Electricity

      When high-voltage charges exist close to semiconductor devices, discharges can cause abnormal operation. In such cases, use anti-static measures or processing to prevent discharges.

   3. Corrosive Gases, Dust, or Oil

      Exposure to corrosive gases or contact with dust or oil may lead to chemical reactions that will adversely affect the device. If you use devices in such conditions, consider ways to prevent such exposure or to protect the devices.

   4. Radiation, Including Cosmic Radiation

      Most devices are not designed for environments involving exposure to radiation or cosmic radiation. Users should provide shielding as appropriate.

   5. Smoke, Flame

      Note: Plastic molded devices are flammable, and therefore should not be used near combustible substances. If devices begin to smoke or burn, there is danger of the release of toxic gases.

Customers considering the use of FUJITSU MICROELECTRONICS products in other special environmental conditions should consult with sales representatives.

■ Notes on handling the device

● Power pins

Because there are multiple V_{CC} and V_{SS} pins, respective pins at the same potential are interconnected to prevent malfunctions such as latch-up. However, you must connect the pins externally to the power supply and ground lines to reduce the electro-magnetic emission levels, to prevent abnormal operation of strobe signals caused by the rise in the ground level, and to conform to the total output current rating. Furthermore, the current supply source should be connected to the V_{CC} and V_{SS} pins of the device at a low impedance.

It is recommended to connect a ceramic bypass capacitor of approximately 0.1 µF as a bypass capacitor between the V_{CC} and V_{SS} pins near this device.

● Crystal oscillation circuit

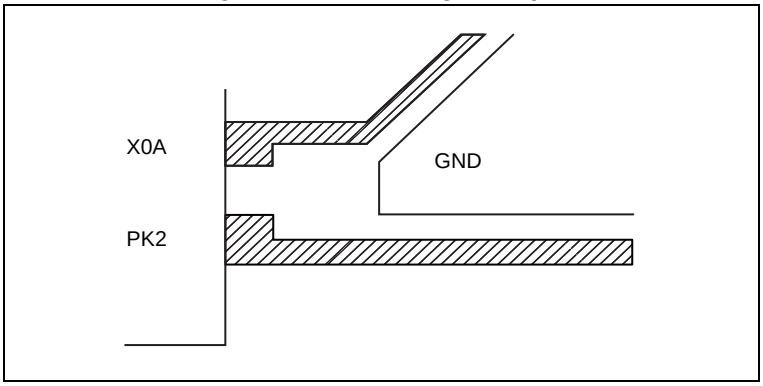
Noise around X0 and X1 pins causes this device to malfunction. Design the PC board so that X0, X1, the crystal oscillator, and the bypass capacitors to the ground are as close as possible.

We also strongly recommend such PC board artwork where pins X0 and X1 are surrounded by grounding because this is expected to ensure stable device operation.

When using 32-KHz oscillation (X0A and X1A), ensure that the changes in PK2 pin input are as small as possible. Perform the processing shown in the following chart so that the wiring to X0A pin and that to PK2 pin cannot run in parallel.

If you do not use 32-KHz oscillation, there is no restriction on the signal at this pin.

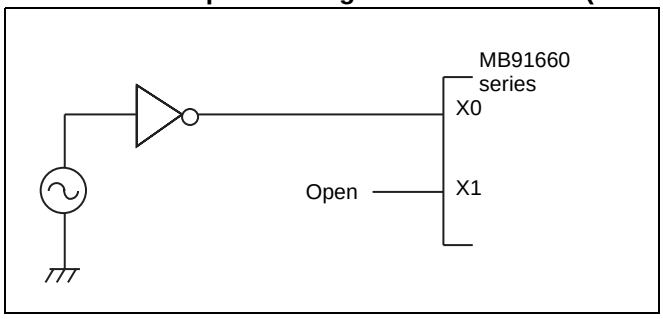
Figure 39.1-1 Wiring example



● Notes on using an external clock

When an external clock is selected, use X0 pin only. X1 pin should be opened.

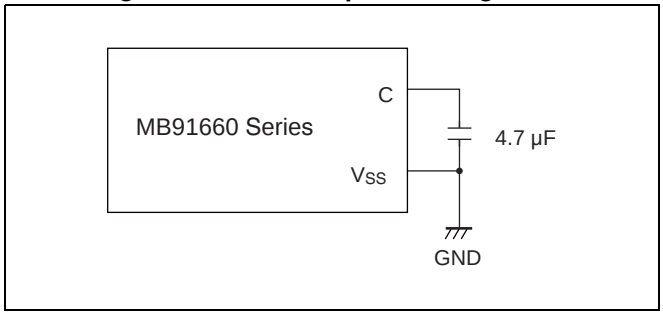
Figure 39.1-2 Example of using an external clock (ordinary)



● C pin

As this series includes an internal regulator, always connect a bypass capacitor of approximately 4.7 µF to the C pin for use by the regulator.

Figure 39.1-3 Example of using C Pin



● MD0 and MD1 (Mode pins)

Connect the mode pins (MD0, MD1) directly to V_{CC} or V_{SS} pins. Design the printed circuit board such that the pull-up/ down resistance stays low, as well as the distance between the mode pins and power supply or GND pins is as short as possible and the connection impedance is low, when the pins are pulled-up/down such as for switching the pin level and rewriting the Flash memory data. It is because of preventing the device erroneously switching to test mode due to noise.

● Power-on sequence

- To guarantee an oscillation stabilization wait time of the internal regulator + oscillation circuit, immediately after power-on, leave the input to pin $\overline{INIT}$ "L" level for a time that is the sum of the regulator voltage stabilization wait time, the oscillator startup time, and the main oscillation stabilization wait time.
- Turn power on/off in the following order.

  power-on: $V_{CC} \rightarrow AV_{CC} \rightarrow AVRH$

  power-off: $AVRH \rightarrow AV_{CC} \rightarrow V_{CC}$
- When canceling the reset state (by changing the level at pin $\overline{INIT}$ from "L" to "H"), ensure that the power is stable.

● Caution on Operations during PLL Clock Mode

On this microcontroller, if in case the crystal oscillator breaks off or an external reference clock input stops while the PLL clock mode is selected, a self-oscillator circuit contained in the PLL may continue its operation at its self-running frequency.

However, Fujitsu will not guarantee results of operations if such failure occurs.

■ Notes on program status registers (PS)

During the execution of a certain instruction, the program status register (PS) is processed in advance. Therefore, the following exception handling operation might cause an interrupt handling routine to encounter a break when the debugger is in use. Alternatively, it might cause the interrupt handling routine to update displayed flags of the program status register (PS).In either case, the device performs processing before and after EIT operation in accordance with the specification because the device is designed to correctly perform re-processing after return from the EIT.

1. When one of events 1 to 3:
   1. A user interrupt is accepted.
   2. Single-step execution is performed.
   3. A break occurs due to a data event or from the emulator menu.

   occurs immediately before the DIV0U/DIV0S instruction, the following operation might be performed:
   - The D0 and D1 flags are updated in advance.
   - The EIT handling routine (user interrupt or emulator) is executed.
   - After return from the EIT, the DIV0U/DIV0S instruction is executed to update the D0 and D1 flags to the same values as in step 1.
2. If the ORCCR/STILM/MOV Ri or PS instruction is executed to enable interrupts when a user interrupt request has been generated, the following operation will result:
   - The program status register (PS) is updated in advance.
   - The EIT processing routine (user interrupt or emulator) is executed.
   - After return from the EIT, the above instruction is executed to update the program status register

(PS) to the same value as in step 1.

■ Debugger-related notes

● Single-step execution of the RETI instruction

In an environment where single-step instruction execution will encounter frequent interrupts, only the appropriate interrupt processing routine for this case is executed repeatedly. As a result, the main routine and programs assigned lower interrupt levels are not executed. (For example, if base timer interrupts are enabled, single-step RETI execution will always cause a break whenever the base timer routine begins.)

When the interrupt handling routine no longer needs to be debugged, disable the pertinent interrupt.

● Break function

If target addresses for hardware breaks (including event breaks) are set to current system stack pointer addresses or set in the area containing stack pointers, the user program will encounter a break after executing one instruction even though it does not contain an actual data access instruction.

To avoid this problem, do not specify (word) access to the area containing system stack pointer addresses as a target of hardware breaks (including event breaks).

● Built-in ROM (Flash memory, mask ROM)

- Notes on using the EVA chip
  - Do not specify any built-in ROM area as a transfer destination specified by the DMA controller (DMAC).
  - If a built-in ROM area is specified as a DMA controller (DMAC) transfer destination, the built-in ROM area might be overwritten when a break occurs during DMAC transfer.
  - It is, however, possible to specify a built-in ROM area as a DMA controller (DMAC) transfer source.

● Operand break

A malfunction could occur if a stack pointer exists in an area that is specified as DSU operand breaks. Do not specify access to the area containing system stack pointer addresses as the target of data event breaks.

APPENDIXES

These appendixes explain the I/O map, a list of registers, the pin state in each CPU state, and a list of instructions for the FR80 family CPUs.

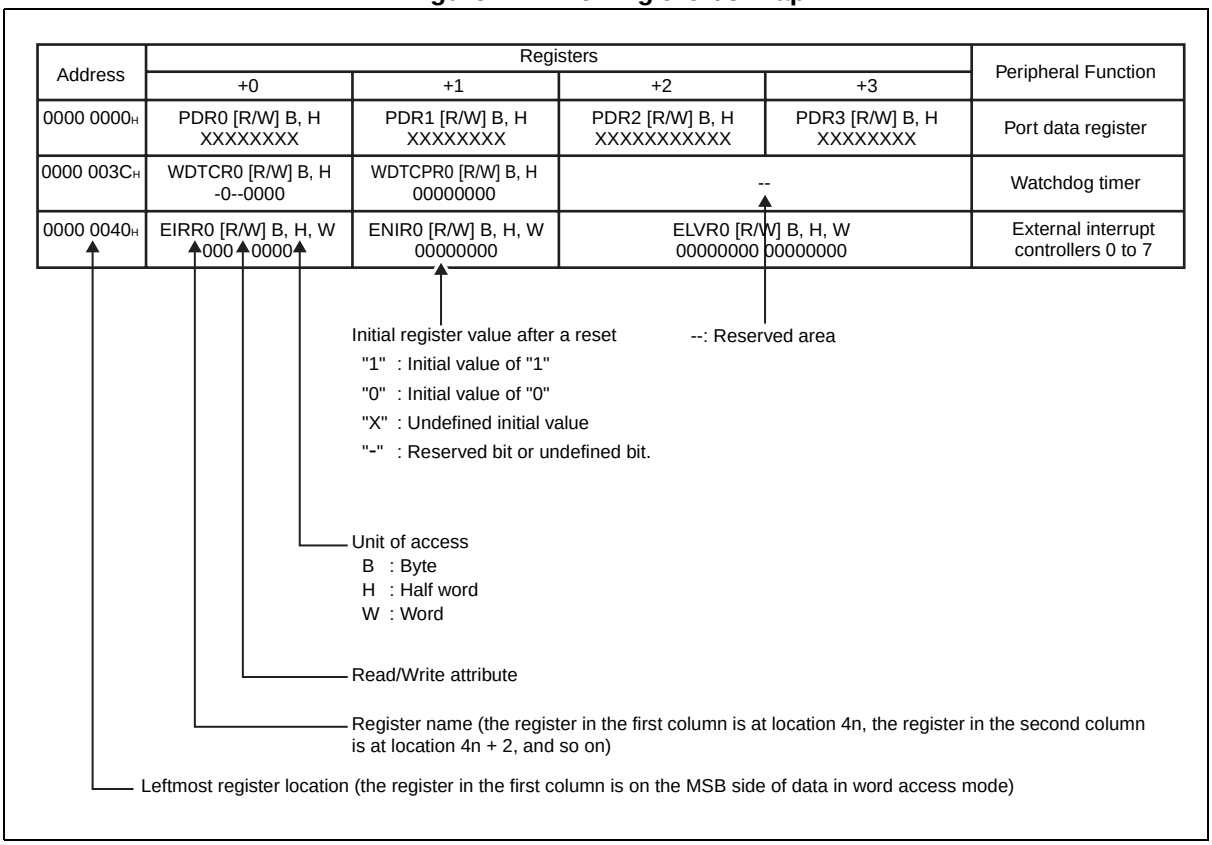
APPENDIX A I/O Map
APPENDIX B List of Registers
APPENDIX C Interrupt Vectors
APPENDIX D Pin State in Each CPU State
APPENDIX E Lists of Instructions

APPENDIX A I/O Map

This appendix outlines the relationship between memory space areas and each register for peripheral functions.

■ Viewing the I/O map

Figure A-1 Viewing the I/O map


Address
Registers
+0
+1
+2
+3
Peripheral Function
0000 0000H
PDR0 [R/W] B, H XXXXXXXX
PDR1 [R/W] B, H XXXXXXXX
PDR2 [R/W] B, H XXXXXXXXXX
PDR3 [R/W] B, H XXXXXXXX
Port data register
0000 003CH
WDTCR0 [R/W] B, H -0--0000
WDTCPR0 [R/W] B, H 00000000
--
Watchdog timer
0000 0040H
EIRR0 [R/W] B, H, W 000 0000
ENIR0 [R/W] B, H, W 00000000
ELVR0 [R/W] B, H, W 00000000 00000000
External interrupt controllers 0 to 7
Initial register value after a reset
"1" : Initial value of "1"
"0" : Initial value of "0"
"X" : Undefined initial value
"-" : Reserved bit or undefined bit.
--: Reserved area
Unit of access
B : Byte
H : Half word
W : Word
Read/Write attribute
Register name (the register in the first column is at location 4n, the register in the second column is at location 4n + 2, and so on)
Leftmost register location (the register in the first column is on the MSB side of data in word access mode)

<Notes>

- Use the following address for data access depending on the access size.
  - Word access: The address is a multiple of 4 (the lowest 2 bits are set to "00_B").
  - Half word access: The address is a multiple of 2 (the lowest bit is set to "0_B").
  - Byte access: --
- Do not access the reserved areas.

Table A-1 I/O map (1 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0000$_H$	PDR0 [R/W] B,H XXXXXXXX	PDR1 [R/W] B,H XXXXXXXX	PDR2 [R/W] B,H XXXXXXXX	PDR3 [R/W] B,H XXXXXXXX	Port data register
0000 0004$_H$	PDR4 [R/W] B,H XXXXXXXX	PDR5 [R/W] B,H XXXXXXXX	PDR6 [R/W] B,H XXXXXXXX	PDR7[R/W] B,H XXXXXXXX	
0000 0008$_H$	PDR8 [R/W] B,H XXXXXXXX	PDR9 [R/W] B,H -- - - - XXX	PDRA [R/W] B XXXXXXXX	—	
0000 000C$_H$	—				
0000 0010$_H$	PDRG [R/W] B,H XXXXXXXX	PDRH [R/W] B,H ----XXXX	—		
0000 0014$_H$	PDRK [R/W] B ----XXXX	—			
0000 0018$_H$ to 0000 001C$_H$	—				
0000 0020$_H$ to 0000 0038$_H$	—				Reserved
0000 003C$_H$	WDTCR0[R/W] B,H - 0- - 0000	WDTCPR0[R/W] B,H 00000000	—		Watchdog timer
0000 0040$_H$	EIRR0[R/W] B,H,W 00000000	ENIR0[R/W] B,H,W 00000000	ELVR0[R/W] B,H,W 00000000 00000000		External interrupt 0 to 7
0000 0044$_H$	DICR [R/W] B - - - - - - - 0	—			Delay interrupt
0000 0048$_H$	TMRLRA0 [R/W] H XXXXXXXX XXXXXXXX		TMR0 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.0
0000 004C$_H$	—		TMCSR0 [R/W] H - - 000000 - - 000000		
0000 0050$_H$	TMRLRA1 [R/W] H XXXXXXXX XXXXXXXX		TMR1 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.1
0000 0054$_H$	—		TMCSR1 [R/W] H - - 000000 - - 000000		
0000 0058$_H$	TMRLRA2 [R/W] H XXXXXXXX XXXXXXXX		TMR2 [R] H XXXXXXXX XXXXXXXX		16-bit reload timer ch.2
0000 005C$_H$	—		TMCSR2 [R/W] H - - 000000 - - 000000		

Table A-1 I/O map (2 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
$0000\ 0060_H$	SCR0 [R/W] B,H,W 0 - - 00000	SMR0 [R/W] B,H,W 000 - 0000	SSR0 [R,R/W] B,H,W 0 - 000011	ESCR0 [R/W] B,H,W - 0000000	Multi-function serial interface ch.0
$0000\ 0064_H$	RDR0[R]/TDR0[W] B,H,W*1 - - - - - - - 0 00000000		BGR10[R/W]H,W 00000000	BGR00[R/W] H,W 00000000	
$0000\ 0068_H$	SCR1[R/W] IBCR1[R,R/W] B,H,W*2 0 - - 00000	SMR1 [R/W] B,H,W 000 - 0000	SSR1 [R,R/W] B,H,W 0 - 000011	ESCR1[R/W] IBSR1[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.1
$0000\ 006C_H$	RDR1[R]/TDR1[W] B,H,W*1 - - - - - - - 0 00000000		BGR11[R/W] H,W 00000000	BGR01[R/W] H,W 00000000	
$0000\ 0070_H$	ISMK1 [R/W] B,H*2 - - - - - - - -	ISBA1 [R/W] B,H*2 - - - - - - - -	—		
$0000\ 0074_H$	SCR2[R/W] IBCR2[R,R/W] B,H,W*2 0 - - 00000	SMR2 [R/W] B,H,W 000 - 0000	SSR2 [R,R/W] B,H,W 0 - 000011	ESCR2[R/W] IBSR2 [R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.2
$0000\ 0078_H$	RDR2[R]/TDR2[W] B,H,W*1 - - - - - - - 0 00000000		BGR12[R/W] H,W 00000000	BGR02[R/W] H,W 00000000	
$0000\ 007C_H$	ISMK2 [R/W] B,H*2 - - - - - - - -	ISBA2 [R/W] B,H*2 - - - - - - - -	—		
$0000\ 0080_H$	SCR3[R/W] IBCR3[R,R/W] B,H,W*2 0 - - 00000	SMR3 [R/W] B,H,W 000 - 0000	SSR3 [R,R/W] B,H,W 0 - 000011	ESCR3[R/W] IBSR3[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.3
$0000\ 0084_H$	RDR3[R]/TDR3[W] B,H,W*1 - - - - - - - 0 00000000		BGR13[R/W] H,W 00000000	BGR03[R/W] H,W 00000000	
$0000\ 0088_H$	ISMK3 [R/W] B,H*2 - - - - - - - -	ISBA3 [R/W] B,H*2 - - - - - - - -	—		
$0000\ 008C_H$	SCR4[R/W] IBCR4[R,R/W] B,H,W*2 0 - - 00000	SMR4 [R/W] B,H,W 000 - 0000	SSR4 [R,R/W] B,H,W 0 - 000011	ESCR4[R/W] IBSR4[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.4
$0000\ 0090_H$	RDR4[R]/TDR4[W] B,H,W*1 - - - - - - - 0 00000000		BGR14[R/W] H,W 00000000	BGR04[R/W] H,W 00000000	
$0000\ 0094_H$	ISMK4 [R/W] B,H*2 - - - - - - - -	ISBA4 [R/W] B,H*2 - - - - - - - -	—		

MB91660 Series

Table A-1 I/O map (3 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0098$_H$	SCR5[R/W] IBCR5[R,R/W] B,H,W*2 0 - - 00000	SMR5 [R/W] B,H,W 000 - 0000	SSR5 [R,R/W] B,H,W 0- 000011	ESCR5[R/W] IBSR5[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.5
0000 009C$_H$	RDR5[R]/TDR5[W] B,H,W*1 - - - - - - - 0 00000000		BGR15 [R/W] H,W 00000000	BGR05 [R/W] H,W 00000000	
0000 00A0$_H$	ISMK5 [R/W] B,H*2 - - - - - - - -	ISBA5 [R/W] B,H*2 - - - - - - - -	—		
0000 00A4$_H$	SCR6[R/W] IBCR6[R,R/W] B,H,W*2 0 - - 00000	SMR6 [R/W] B,H,W 000 - 0000	SSR6 [R,R/W] B,H,W 0 - 000011	ESCR6[R/W] IBSR6[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.6
0000 00A8$_H$	RDR6[R]/TDR6[W] B,H,W*1 - - - - - - - 0 00000000		BGR16 [R/W] H,W 00000000	BGR06 [R/W] H,W 00000000	
0000 00AC$_H$	ISMK6 [R/W] B,H*2 - - - - - - - -	ISBA6 [R/W] B,H*2 - - - - - - - -	—		
0000 00B0$_H$	SCR7 [R/W] IBCR7[R,R/W] B,H,W*2 0 - - 00000	SMR7 [R/W] B,H,W 000 - 0000	SSR7 [R,R/W] B,H,W 0 - 000011	ESCR7 [R/W] IBSR7 [R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.7
0000 00B4$_H$	RDR7[R]/TDR7[W] B,H,W*1 - - - - - - - 0 00000000		BGR17 [R/W] H,W 00000000	BGR07 [R/W] H,W 00000000	
0000 00B8$_H$	ISMK7 [R/W] B,H*2 - - - - - - - -	ISBA7 [R/W] B,H*2 - - - - - - - -	—		
0000 00BC$_H$	—				Reserved
0000 00C0$_H$	RDRM0 [R]/ TDRM0[W] B,H,W 00000000	RDRM1 [R]/ TDRM1[W] B,H,W 00000000	RDRM2 [R]/ TDRM2[W] B,H,W 00000000	RDRM3 [R]/ TDRM3[W] B,H,W 00000000	Multi-function serial interface data register (mirror)
0000 00C4$_H$	RDRM4 [R]/ TDRM4[W] B,H,W 00000000	RDRM5 [R]/ TDRM5[W] B,H,W 00000000	RDRM6 [R]/ TDRM6[W] B,H,W 00000000	RDRM7 [R]/ TDRM7[W] B,H,W 00000000	
0000 00C8$_H$	SSEL0123 [R/W] B - - - - - - 00	—	SSEL4567 [R/W] B - - - - - - 00	—	Multi-function serial interface serial clock selection
0000 00CC$_H$	—				Reserved

Table A-1 I/O map (4 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 00D0$_H$	SCR8 [R/W] IBCR8 [R,R/W] B,H,W*2 0 - - 00000	SMR8 [R/W] B,H,W 000 - 0000	SSR8 [R,R/W] B,H,W 0 - 000011	ESCR8 [R/W] IBSR8 [R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.8 (FIFO)
0000 00D4$_H$	RDR8[R]/TDR8[W] B,H,W*1 - - - - - - - 0 00000000		BGR18 [R/W] H,W 00000000	BGR08 [R,R/W] H,W 00000000	
0000 00D8$_H$	ISMK8 [R/W] B,H*2 - - - - - - - -	ISBA8 [R/W] B,H*2 - - - - - - - -	—		
0000 00DC$_H$	FCR18 [R/W] B,H,W - - - 00100	FCR08 [R,R/W] B,H,W - 0000000	FBYTE28 [R/W] B,H,W 00000000	FBYTE18 [R/W] B,H,W 00000000	
0000 00E0$_H$	SCR9 [R/W] IBCR9 [R,R/W] B,H,W*2 0- - 00000	SMR9 [R/W] B,H,W 000 - 0000	SSR9 [R,R/W] B,H,W 0 - 000011	ESCR9 [R/W] IBSR9[R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.9 (FIFO)
0000 00E4$_H$	RDR9[R]/TDR9[W] B,H,W*1 - - - - - - - 0 00000000		BGR19 [R/W] H,W 00000000	BGR09 [R/W] H,W 00000000	
0000 00E8$_H$	ISMK9 [R/W] B,H*2 - - - - - - - -	ISBA9 [R/W] B,H*2 - - - - - - - -	—		
0000 00EC$_H$	FCR19 [R/W] B,H,W - - - 00100	FCR09 [R,R/W] B,H,W - 0000000	FBYTE29 [R/W] B,H,W 00000000	FBYTE19 [R/W] B,H,W 00000000	
0000 00F0$_H$	SCR10 [R/W] IBCR10 [R,R/W] B,H,W*2 0- - 00000	SMR10 [R/W] B,H,W 000- 0000	SSR10 [R,R/W] B,H,W 0- 000011	ESCR10 [R/W] IBSR10 [R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.10 (FIFO)
0000 00F4$_H$	RDR10[R]/TDR10[W] B,H,W*1 - - - - - - - 0 00000000		BGR110 [R/W] H,W 00000000	BGR010 [R/W] H,W 00000000	
0000 00F8$_H$	ISMK10 [R/W] B,H*2 - - - - - - - -	ISBA10 [R/W] B,H*2 - - - - - - - -	—		
0000 00FC$_H$	FCR110 [R/W] B,H,W - - - 00100	FCR010 [R,R/W] B,H,W - 0000000	FBYTE210 [R/W] B,H,W 00000000	FBYTE110 [R/W] B,H,W 00000000	

MB91660 Series

Table A-1 I/O map (5 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0100$_H$	SCR11 [R/W] IBCR11 [R,R/W] B,H,W*2 0- - 00000	SMR11 [R/W] B,H,W 000- 0000	SSR11 [R,R/W] B,H,W 0- 000011	ESCR11 [R/W] IBSR11 [R,R/W] B,H,W*2 - 0000000	Multi-function serial interface ch.11 (FIFO)
0000 0104$_H$	RDR11[R]/TDR11[W] B,H,W*1 - - - - - - - 0 00000000		BGR111 [R/W] H,W 00000000	BGR011 [R/W] H,W 00000000	
0000 0108$_H$	ISMK11 [R/W] B,H*2 - - - - - - - -	ISBA11 [R/W] B,H*2 - - - - - - - -	—		
0000 010C$_H$	FCR111 [R/W] B,H,W - - - 00100	FCR011 [R,R/W] B,H,W - 0000000	FBYTE211 [R/W] B,H,W 00000000	FBYTE111 [R/W] B,H,W 00000000	
0000 0110$_H$	EIRR1[R/W] B,H,W 00000000	ENIR1[R/W] B,H,W 00000000	ELVR1[R/W] B,H,W 00000000 00000000		External interrupt 8 to 15
0000 0114$_H$	EIRR2[R/W] B,H,W 00000000	ENIR2[R/W] B,H,W 00000000	ELVR2[R/W] B,H,W 00000000 00000000		External interrupt 16 to 23
0000 0118$_H$	EIRR3[R/W] B,H,W 00000000	ENIR3[R/W] B,H,W 00000000	ELVR3[R/W] B,H,W 00000000 00000000		External interrupt 24 to 31
0000 011C$_H$	—				Reserved
0000 0120$_H$	ADCR0[R/W] B,H 000- 0000	ADSR0[R,R/W] B,H 00- - - 000	—		A/D converter
0000 0124$_H$	SCCR0[R,R/W] B,H 1000- 000	SFNS0[R/W] B,H - - - - 0000	SCFD0[R] B,H XXXXXXXX XX- XXXXX		
0000 0128$_H$	—	SCIS20[R/W] B 00000000	SCIS10[R/W] B,H 00000000	SCIS00[R/W] B,H 00000000	
0000 012C$_H$	PCCR0[R,R/W] B,H 1000- 000	PFNS0[R/W] B,H - - - - - - 00	PCFD0[R] B,H XXXXXXXX XXXXXXXX		
0000 0130$_H$	PCIS0[R/W] B 00000000	—	CMPD0[R/W] B,H 00000000	CMPCR0[R/W] B,H 00000000	
0000 0134$_H$	—	ADSS20[R/W] B 00000000	ADSS10[R/W] B,H 00000000	ADSS00[R/W] B,H 00000000	
0000 0138$_H$	ADST00[R/W] B,H 00100000	ADST10[R/W] B,H 00100000	ADCT0[R/W] B - - - - - 111	—	
0000 013C$_H$	—				Reserved

Table A-1 I/O map (6 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0140$_H$	BT0TMR[R]H 00000000 00000000		BT0TMCR[R/W] B,H -0000000 00000000		Base timer ch.0
0000 0144$_H$	—	BT0STC[R/W]B 0000-000	—		
0000 0148$_H$	BT0PCSR/BT0PRLL[R/W]H XXXXXXXX XXXXXXXX		BT0PDUT/BT0PRLH/BT0DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 014C$_H$	—				
0000 0150$_H$	BT1TMR[R]H 00000000 00000000		BT1TMCR[R/W] B,H -0000000 00000000		Base timer ch.1
0000 0154$_H$	—	BT1STC[R/W]B 0000-000	—		
0000 0158$_H$	BT1PCSR/BT1PRLL[R/W]H XXXXXXXX XXXXXXXX		BT1PDUT/BT1PRLH/BT1DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 015C$_H$	—				
0000 0160$_H$	BT2TMR[R]H 00000000 00000000		BT2TMCR [R/W] B,H -0000000 00000000		Base timer ch.2
0000 0164$_H$	—	BT2STC[R/W]B 0000-000	—		
0000 0168$_H$	BT2PCSR/BT2PRLL[R/W]H XXXXXXXX XXXXXXXX		BT2PDUT/BT2PRLH/BT2DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 016C$_H$	—				
0000 0170$_H$	BT3TMR[R]H 00000000 00000000		BT3TMCR[R/W] B,H -0000000 00000000		Base timer ch.3
0000 0174$_H$	—	BT3STC[R/W]B 0000-000	—		
0000 0178$_H$	BT3PCSR/BT3PRLL[R/W]H XXXXXXXX XXXXXXXX		BT3PDUT/BT3PRLH/BT3DTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 017C$_H$	BTSEL0123 [R/W] B 00000000	—			
0000 0180$_H$	DACR0[R/W] B,H,W - - - - - - - 0	DADR0[R/W] B,H,W XXXXXXXX	DACR1[R/W] B,H,W - - - - - - - 0	DADR1[R/W] B,H,W XXXXXXXX	D/A converter
0000 0184$_H$	DACR2[R/W] B,H - - - - - - - 0	DADR2[R/W] B,H XXXXXXXX	—		
0000 0188$_H$ to 0000 018C$_H$	—				

MB91660 Series

Table A-1 I/O map (7 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0190$_H$ to 0000 01A8$_H$	—				Reserved
0000 01AC$_H$	ADCHE [R/W] B,H,W -------- 11111111 11111111 11111111				A/D channel enable
0000 01B0$_H$	IRPR0H [R] B 000- - - - -	—	IRPR1H [R] B,H 000- 000-	IRPR1L [R] B,H 000- 000-	Interrupt request batch read function
0000 01B4$_H$	IRPR2H [R] B,H,W 0000- - - -	IRPR2L [R] B,H,W 000- - - - -	IRPR3H [R] B,H,W 0000- - - -	IRPR3L [R] B,H,W 00000 - - -	
0000 01B8$_H$	IRPR4H [R] B,H,W 0000- - - -	IRPR4L [R] B,H,W 000000- -	IRPR5H [R] B,H,W 0000- - - -	IRPR5L [R] B,H,W 0000- - - -	
0000 01BC$_H$	IRPR6H [R] B,H,W 0000- - - -	IRPR6L [R] B,H,W 0000- - - -	IRPR7H [R] B,H,W 0000- 0- -	IRPR7L [R] B,H,W 0000- - - -	
0000 01C0$_H$	RCRH0 [W] H,W 00000000	RCRL0 [W] B,H,W 00000000	UDCRH0 [R] H,W 00000000	UDCRL0 [R] B,H,W 00000000	Up/down counter ch.0
0000 01C4$_H$	CCR0 [R, R/W] B,H 00000000 -0001000		—	CSR0 [R,R/W] B 00000000	
0000 01C8$_H$	—				
0000 01CC$_H$	—				Reserved
0000 01D0$_H$	RCRH1 [W] H,W 00000000	RCRL1 [W] B,H,W 00000000	UDCRH1 [R] H,W 00000000	UDCRL1 [R] B,H,W 00000000	Up/down counter ch.1
0000 01D4$_H$	CCR1 [R,R/W] B,H 00000000 -0001000		—	CSR1 [R,R/W] B 00000000	
0000 01D8$_H$	—				
0000 01DC$_H$	—				Reserved
0000 01E0$_H$	RCRH2 [W] H,W 00000000	RCRL2 [W] B,H,W 00000000	UDCRH2 [R] H,W 00000000	UDCRL2 [R] B,H,W 00000000	Up/down counter ch.2
0000 01E4$_H$	CCR2 [R, R/W] B,H 00000000 -0001000		—	CSR2 [R,R/W] B 00000000	
0000 01E8$_H$	—				
0000 01EC$_H$	—				Reserved
0000 01F0$_H$	RCRH3 [W] H,W 00000000	RCRL3 [W] B,H,W 00000000	UDCRH3 [R] H,W 00000000	UDCRL3 [R] B,H,W 00000000	Up/down counter ch.3
0000 01F4$_H$	CCR3 [R,R/W] B,H 00000000 -0001000		—	CSR3 [R,R/W] B 00000000	
0000 01F8$_H$	—				
0000 01FC$_H$	—				Reserved

Table A-1 I/O map (8 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0200$_H$	CPCLR0 [R/W] W 11111111 11111111 11111111 11111111				32-bit Free-run timer ch.0
0000 0204$_H$	TCDT0 [R/W] W 00000000 00000000 00000000 00000000				
0000 0208$_H$	TCCSH0 [R/W] B,H 0- - - - - 00	TCCSL0 [R/W] B,H - 1- 00000	—		
0000 020C$_H$	IPCP0 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				32-bit Input capture ch.0 to ch.3
0000 0210$_H$	IPCP1 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0214$_H$	IPCP2 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0218$_H$	IPCP3 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 021C$_H$	—	ICS01 [R/W] B 00000000	—	ICS23 [R/W] B 00000000	
0000 0220$_H$	IPCP4 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				32-bit Input capture ch.4 to ch.7
0000 0224$_H$	IPCP5 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0228$_H$	IPCP6 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 022C$_H$	IPCP7 [R] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0230$_H$	—	ICS45 [R/W] B 00000000	—	ICS67 [R/W] B 00000000	
0000 0234$_H$	OCCP0 [R/W] W 00000000 00000000 00000000 00000000				32-bit Output compare ch.0 to ch.3
0000 0238$_H$	OCCP1 [R/W] W 00000000 00000000 00000000 00000000				
0000 023C$_H$	OCCP2 [R/W] W 00000000 00000000 00000000 00000000				
0000 0240$_H$	OCCP3 [R/W] W 00000000 00000000 00000000 00000000				
0000 0244$_H$	OCSH1 [R/W] B,H,W - - - 0- - 00	OCSL0 [R/W] B,H,W 0000- - 00	OCSH3 [R/W] B,H,W - - - 0- - 00	OCSL2 [R/W] B,H,W 0000- - 00	

MB91660 Series

Table A-1 I/O map (9 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0248$_H$	OCCP4 [R/W] W 00000000 00000000 00000000 00000000				32-bit Output compare ch.4 to ch.7
0000 024C$_H$	OCCP5 [R/W] W 00000000 00000000 00000000 00000000				
0000 0250$_H$	OCCP6 [R/W] W 00000000 00000000 00000000 00000000				
0000 0254$_H$	OCCP7 [R/W] W 00000000 00000000 00000000 00000000				
0000 0258$_H$	OCSH5 [R/W] B,H,W - - - 0- - 00	OCSL4 [R/W] B,H,W 0000- - 00	OCSH7 [R/W] B,H,W - - - 0- - 00	OCSL6 [R/W] B,H,W 0000- - 00	
0000 025C$_H$	FRTSEL [R/W] B - - - - - - 00	—			Free-run timer selector
0000 0260$_H$	CPCLR1 [R/W] W 11111111 11111111 11111111 11111111				32-bit Free-run timer ch.1
0000 0264$_H$	TCDT1 [R/W] W 00000000 00000000 00000000 00000000				
0000 0268$_H$	TCCSH1 [R/W] B,H 0- - - - - 00	TCCSL1 [R/W] B,H - 1- 00000	—		
0000 026C$_H$ to 0000 031C$_H$	—				Reserved
0000 0320$_H$	FCTLR[R/W] H - 0- - 1011 - - - - - - - -		—	FSTR[R] B - - - - - - - 0	Flash memory control
0000 0324$_H$ to 0000 0334$_H$	—				Reserved
0000 0338$_H$	—		WREN[R/W] B,H 00000000 00000000		Wild register
0000 033C$_H$	—				
0000 0340$_H$ to 0000 037C$_H$	—				Reserved

Table A-1 I/O map (10 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0380$_H$	WRAR00[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				Wild register
0000 0384$_H$	WRDR00[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0388$_H$	WRAR01[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 038C$_H$	WRDR01[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0390$_H$	WRAR02[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 0394$_H$	WRDR02[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0398$_H$	WRAR03[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 039C$_H$	WRDR03[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03A0$_H$	WRAR04[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03A4$_H$	WRDR04[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03A8$_H$	WRAR05[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03AC$_H$	WRDR05[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03B0$_H$	WRAR06[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03B4$_H$	WRDR06[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03B8$_H$	WRAR07[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03BC$_H$	WRDR07[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03C0$_H$	WRAR08[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03C4$_H$	WRDR08[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

MB91660 Series

Table A-1 I/O map (11 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 03C8$_H$	WRAR09[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				Wild register
0000 03CC$_H$	WRDR09[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03D0$_H$	WRAR10[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03D4$_H$	WRDR10[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03D8$_H$	WRAR11[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03DC$_H$	WRDR11[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03E0$_H$	WRAR12[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03E4$_H$	WRDR12[R/W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03E8$_H$	WRAR13[R/W] - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03EC$_H$	WRDR13[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03F0$_H$	WRAR14[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03F4$_H$	WRDR14[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 03F8$_H$	WRAR15[R/W] W - - - - - - - - - - XXXXXX XXXXXXXX XXXXXX- -				
0000 03FC$_H$	WRDR15[R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0400$_H$	DDR0 [R/W] B,H 00000000	DDR1 [R/W] B,H 00000000	DDR2 [R/W] B,H 00000000	DDR3 [R/W] B,H 00000000	Data direction register
0000 0404$_H$	DDR4 [R/W] B,H 00000000	DDR5 [R/W] B,H 00000000	DDR6 [R/W] B,H 00000000	DDR7[R/W] B,H 00000000	
0000 0408$_H$	DDR8 [R/W] B,H 00000000	DDR9 [R/W] B,H - - - - - 000	DDRA [R/W] B 00000000	—	
0000 040C$_H$	—				
0000 0410$_H$	DDRG [R/W] B,H 00000000	DDRH [R/W] B,H - - - - 0000	—		
0000 0414$_H$	DDRK [R/W] B - - - - 0000	—			
0000 0418$_H$ to 0000 041C$_H$	—				Data direction register

Table A-1 I/O map (12 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0420$_H$	PCR0 [R/W] B,H 00000000	PCR1 [R/W] B,H 00000000	—		Pull-up control register
0000 0424$_H$	—	PCR5 [R/W] B 00000000	PCR6 [R/W] B,H 00000000	PCR7[R/W] B,H 00000000	
0000 0428$_H$	PCR8 [R/W] B,H 00000000	PCR9 [R/W] B,H - - - - - 000	PCRA [R/W] B 00000000	—	
0000 042C$_H$ to 0000 0430$_H$	—				
0000 0434$_H$	PCRK [R/W] B - - - - 00- -	—			
0000 0438$_H$ to 0000 043C$_H$	—				
0000 0440$_H$	ICR00 [R,R/W] B,H,W - - - 11111	ICR01 [R,R/W] B,H,W - - - 11111	ICR02 [R,R/W] B,H,W - - - 11111	ICR03 [R,R/W] B,H,W - - - 11111	Interrupt control
0000 0444$_H$	ICR04 [R,R/W] B,H,W - - - 11111	ICR05 [R,R/W] B,H,W - - - 11111	ICR06 [R,R/W] B,H,W - - - 11111	ICR07 [R,R/W] B,H,W - - - 11111	
0000 0448$_H$	ICR08 [R,R/W] B,H,W - - - 11111	ICR09 [R,R/W] B,H,W - - - 11111	ICR10 [R,R/W] B,H,W - - - 11111	ICR11 [R,R/W] B,H,W - - - 11111	
0000 044C$_H$	ICR12 [R,R/W] B,H,W - - - 11111	ICR13 [R,R/W] B,H,W - - - 11111	ICR14 [R,R/W] B,H,W - - - 11111	ICR15 [R,R/W] B,H,W - - - 11111	
0000 0450$_H$	ICR16 [R,R/W] B,H,W - - - 11111	ICR17 [R,R/W] B,H,W - - - 11111	ICR18 [R,R/W] B,H,W - - - 11111	ICR19 [R,R/W] B,H,W - - - 11111	
0000 0454$_H$	ICR20 [R,R/W] B,H,W - - - 11111	ICR21 [R,R/W] B,H,W - - - 11111	ICR22 [R,R/W] B,H,W - - - 11111	ICR23 [R,R/W] B,H,W - - - 11111	
0000 0458$_H$	ICR24 [R,R/W] B,H,W - - - 11111	ICR25 [R,R/W] B,H,W - - - 11111	ICR26 [R,R/W] B,H,W - - - 11111	ICR27 [R,R/W] B,H,W - - - 11111	
0000 045C$_H$	ICR28 [R,R/W] B,H,W - - - 11111	ICR29 [R,R/W] B,H,W - - - 11111	ICR30 [R,R/W] B,H,W - - - 11111	ICR31 [R,R/W] B,H,W - - - 11111	

MB91660 Series

Table A-1 I/O map (13 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0460$_H$	ICR32 [R,R/W] B,H,W - - - 11111	ICR33 [R,R/W] B,H,W - - - 11111	ICR34 [R,R/W] B,H,W - - - 11111	ICR35 [R,R/W] B,H,W - - - 11111	Interrupt control
0000 0464$_H$	ICR36 [R,R/W] B,H,W - - - 11111	ICR37 [R,R/W] B,H,W - - - 11111	ICR38 [R,R/W] B,H,W - - - 11111	ICR39 [R,R/W] B,H,W - - - 11111	
0000 0468$_H$	ICR40 [R,R/W] B,H,W - - - 11111	ICR41 [R,R/W] B,H,W - - - 11111	ICR42 [R,R/W] B,H,W - - - 11111	ICR43 [R,R/W] B,H,W - - - 11111	
0000 046C$_H$	ICR44 [R,R/W] B,H,W - - - 11111	ICR45 [R,R/W] B,H,W - - - 11111	ICR46 [R,R/W] B,H,W - - - 11111	ICR47 [R,R/W] B,H,W - - - 11111	
0000 0470$_H$ to 0000 047C$_H$	—				Reserved
0000 0480$_H$	RSTRR [R] B,H,W 11-X- - - X*3	RSTCR [R/W] B,H,W 000- - - - 0	STBCR [R/W] B,H,W 0000- - 11	SLPRR [R/W] B,H,W 00000000	Reset control/Power consumption control
0000 0484$_H$	—				
0000 0488$_H$	DIVR0 [R/W] B,H 000- - - - -	DIVR1 [R/W] B,H 0001- - - -	DIVR2 [R/W] B 0011- - - -	—	Clock division control
0000 048C$_H$	—				
0000 0490$_H$	IORR0 [R/W] B,H,W - 0000000	IORR1 [R/W] B,H,W - 0000000	IORR2 [R/W] B,H,W - 0000000	IORR3 [R/W] B,H,W - 0000000	Peripheral DMA transmission request control
0000 0494$_H$	IORR4 [R/W] B,H,W - 0000000	IORR5 [R/W] B,H,W - 0000000	IORR6 [R/W] B,H,W - 0000000	IORR7 [R/W] B,H,W - 0000000	
0000 0498$_H$ to 0000 049C$_H$	—				
0000 04A0$_H$	PFR0 [R/W] B,H 00000000	PFR1 [R/W] B,H 00000000	PFR2 [R/W] B,H 00000000	PFR3 [R/W] B,H 00000000	Port function register
0000 04A4$_H$	PFR4 [R/W] B,H 00000000	PFR5 [R/W] B,H 00000000	PFR6 [R/W] B,H 00- 00- 0-	PFR7[R/W] B,H 00000000	
0000 04A8$_H$	PFR8 [R/W] B 00000000	—	PFRA [R/W] B 00- 00000	—	
0000 04AC$_H$	—				
0000 04B0$_H$	PFRG [R/W] B,H - 000- 000	PFRH [R/W] B,H - - - - - 0- 0	—		
0000 04B4$_H$	—				

Table A-1 I/O map (14 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 04B8$_H$	EPFR0 [R/W] B,H - - 000000	EPFR1 [R/W] B,H - - 000000	EPFR2 [R/W] B,H - - 000000	EPFR3 [R/W] B,H - - 000000	Extended port function register
0000 04BC$_H$	EPFR4 [R/W] B,H 00000000	EPFR5 [R/W] B,H 00000000	EPFR6 [R/W] B,H 00000000	EPFR7 [R/W] B,H - - - 00000	
0000 04C0$_H$	EPFR8 [R/W] B,H - - - 00000	EPFR9 [R/W] B,H - - - 00000	EPFR10 [R/W] B,H - - - 00000	EPFR11 [R/W] B,H - - - 00000	
0000 04C4$_H$	EPFR12 [R/W] B,H - - - 00000	EPFR13 [R/W] B,H - - - 00000	EPFR14 [R/W] B,H - - - 00000	EPFR15 [R/W] B,H - - - 00000	
0000 04C8$_H$	EPFR16 [R/W] B,H - - - 00000	EPFR17 [R/W] B,H - - - 00000	EPFR18 [R/W] B,H 00000000	EPFR19 [R/W] B,H - - - - 0001	
0000 04CC$_H$	EPFR20 [R/W] B,H - - 000000	EPFR21 [R/W] B,H - - 000000	EPFR22 [R/W] B,H - - 000000	EPFR23 [R/W] B,H - - 000000	
0000 04D0$_H$	EPFR24 [R/W] B,H - - 000000	EPFR25 [R/W] B,H - - 000000	EPFR26 [R/W] B,H - - 000000	EPFR27 [R/W] B,H - - 000000	
0000 04D4$_H$	EPFR28 [R/W] B,H 00000000	EPFR29 [R/W] B,H 00000000	EPFR30 [R/W] B,H - - - - 0000	EPFR31 [R/W] B,H - 0000000	
0000 04D8$_H$	EPFR32 [R/W] B,H 00000000	EPFR33 [R/W] B,H - - 000000	EPFR34 [R/W] B,H - 0000000	EPFR35 [R/W] B,H - - - - - - 00	
0000 04DC$_H$	—				
0000 04E0$_H$ to 0000 04EC$_H$	—				Reserved
0000 04F0$_H$	ICSEL0[R/W] B,H,W - - - - - 000	ICSEL1[R/W] B,H,W - - - - - 000	ICSEL2[R/W] B,H,W - - - - - 000	ICSEL3[R/W] B,H,W - - - - - 000	DMA start request clear select function
0000 04F4$_H$	ICSEL4[R/W] B,H,W - - - - - - 00	ICSEL5[R/W] B,H,W - - - - - 000	ICSEL6[R/W] B,H,W - - - - - - 00	ICSEL7[R/W] B,H,W - - - - - - - 0	
0000 04F8$_H$	ICSEL8[R/W] B,H,W - - - - - - 00	ICSEL9[R/W] B,H,W - - - - - 000	ICSEL10[R/W] B,H,W - - - - 0000	ICSEL11[R/W] B,H,W - - - - 0000	
0000 04FC$_H$	ICSEL12[R/W] B,H - - - - 0000	ICSEL13[R/W] B,H - - - - - 0-0	ICSEL14[R/W] B - - - - - - 00	—	
0000 0500$_H$ to 0000 050C$_H$	—				Reserved

MB91660 Series

Table A-1 I/O map (15 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0510$_H$	CSELR [R/W] B,H,W 001- - - 00	CMONR [R] B,H,W 001- - - 00	MTMCR [R/W] B,H,W 00001111	STMCR [R/W] B,H,W 0000- 111	Clock generation/ Main timer/ Sub timer
0000 0514$_H$	PLLCR [R/W] B,H - - 000000 11110000		CSTBR [R/W] B - 0000000	—	
0000 0518$_H$	WCRD [R] B,H - - 000000	WCRL [R/W] B,H - - 000000	WCCR [R,R/W] B 00- - 0000	—	Clock counter
0000 051C$_H$	UCCR [R/W] B - - - - - 001	—			USB clock generation
0000 0520$_H$ to 0000 05FC$_H$	—				Reserved
0000 0600$_H$	ASR0 [R/W] W 00000000 00000000 - - - - - - - - 1111- 001				External bus I/F
0000 0604$_H$	ASR1 [R/W] W XXXXXXXX XXXXXXXX - - - - - - - - XXXX- XX0				
0000 0608$_H$	ASR2 [R/W] W XXXXXXXX XXXXXXXX - - - - - - - - XXXX- XX0				
0000 060C$_H$	ASR3 [R/W] W XXXXXXXX XXXXXXXX - - - - - - - - XXXX- XX0				
0000 0610$_H$ to 0000 063C$_H$	—				
0000 0640$_H$	ACR0[R/W] W - 00- - 00- 0				
0000 0644$_H$	ACR1[R/W] W - XX- - XX- X				
0000 0648$_H$	ACR2[R/W] W - XX- - XX- X				
0000 064C$_H$	ACR3[R/W] W - XX- - XX- X				
0000 0650$_H$ to 0000 067C$_H$	—				
0000 0680$_H$	AWR0 [R/W] W - - - - 1111 00000000 11110000 00000- 0-				

Table A-1 I/O map (16 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0684$_H$	AWR1 [R/W] W - - - - XXXX XXXXXXXX XXXXXXXX XXXXX- X-				External bus I/F
0000 0688$_H$	AWR2 [R/W] W - - - - XXXX XXXXXXXX XXXXXXXX XXXXX- X-				
0000 068C$_H$	AWR3 [R/W] W - - - - XXXX XXXXXXXX XXXXXXXX XXXXX- X-				
0000 0690$_H$ to 0000 06BC$_H$	—				
0000 06C0$_H$	DMAR0 [R/W] W - 0000				
0000 06C4$_H$	DMAR1 [R/W] W - 0000				
0000 06C8$_H$	DMAR2 [R/W] W - 0000				
0000 06CC$_H$	DMAR3 [R/W] W - 0000				
0000 06D0$_H$ to 0000 06FC$_H$	—				Reserved
0000 0700$_H$ to 0000 0BFC$_H$	—				

MB91660 Series

Table A-1 I/O map (17 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0C00$_H$	DCCR0 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				DMAC
0000 0C04$_H$	DCSR0 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR0 [R/W] H 00000000 00000000		
0000 0C08$_H$	DSAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C0C$_H$	DDAR0 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C10$_H$	DCCR1 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C14$_H$	DCSR1 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR1 [R/W] H 00000000 00000000		
0000 0C18$_H$	DSAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C1C$_H$	DDAR1 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C20$_H$	DCCR2 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C24$_H$	DCSR2 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR2 [R/W] H 00000000 00000000		
0000 0C28$_H$	DSAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C2C$_H$	DDAR2 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C30$_H$	DCCR3 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C34$_H$	DCSR3 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR3 [R/W] H 00000000 00000000		
0000 0C38$_H$	DSAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C3C$_H$	DDAR3 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C40$_H$	DCCR4 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C44$_H$	DCSR4 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR4 [R/W] H 00000000 00000000		
0000 0C48$_H$	DSAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C4C$_H$	DDAR4 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				

Table A-1 I/O map (18 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0C50$_H$	DCCR5 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				DMAC
0000 0C54$_H$	DCSR5 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR5 [R/W] H 00000000 00000000		
0000 0C58$_H$	DSAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C5C$_H$	DDAR5 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C60$_H$	DCCR6 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C64$_H$	DCSR6 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR6 [R/W] H 00000000 00000000		
0000 0C68$_H$	DSAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C6C$_H$	DDAR6 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C70$_H$	DCCR7 [R/W] W 0- - - - 000 - - 00- - 00 00000000 0- 000000				
0000 0C74$_H$	DCSR7 [R, R/W] H 0- - - - - - - - - - - - - 000		DTCR7 [R/W] H 00000000 00000000		
0000 0C78$_H$	DSAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C7C$_H$	DDAR7 [R/W] W XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX				
0000 0C80$_H$ to 0000 0DF0$_H$	—				
0000 0DF4$_H$	—			DILVR [R,R/W] B - - - 11111	
0000 0DF8$_H$	DMACR [R/W] W 0- - - - - - - - - - - - - - - - 0- - - - - - - - - - - - - - - -				
0000 0DFC$_H$ to 0000 0F3C$_H$	—				Reserved
0000 0F40$_H$	BT4TMR[R]H 00000000 00000000		BT4TMCR[R/W] B,H -0000000 00000000		Base timer ch.4
0000 0F44$_H$	—	BT4STC[R/W]B 0000-000	—		
0000 0F48$_H$	BT4PCSR/BT4PRLL[R/W]H XXXXXXXX XXXXXXXX		BT4PDUT/BT4PRLH/BT4DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F4C$_H$	—				

MB91660 Series

Table A-1 I/O map (19 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0F50$_H$	BT5TMR[R]H 00000000 00000000		BT5TMCR[R/W] B,H -0000000 00000000		Base timer ch.5
0000 0F54$_H$	—	BT5STC[R/W]B 0000-000	—		
0000 0F58$_H$	BT5PCSR/BT5PRLL[R/W]H XXXXXXXX XXXXXXXX		BT5PDUT/BT5PRLH/BT5DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F5C$_H$	—				
0000 0F60$_H$	BT6TMR[R]H 00000000 00000000		BT6TMCR[R/W] B,H -0000000 00000000		Base timer ch.6
0000 0F64$_H$	—	BT6STC[R/W]B 0000-000	—		
0000 0F68$_H$	BT6PCSR/BT6PRLL[R/W]H XXXXXXXX XXXXXXXX		BT6PDUT/BT6PRLH/BT6DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F6C$_H$	—				
0000 0F70$_H$	BT7TMR[R]H 00000000 00000000		BT7TMCR[R/W] B,H -0000000 00000000		Base timer ch.7
0000 0F74$_H$	—	BT7STC[R/W]B 0000-000	—		
0000 0F78$_H$	BT7PCSR/BT7PRLL[R/W]H XXXXXXXX XXXXXXXX		BT7PDUT/BT7PRLH/BT7DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F7C$_H$	BTSEL4567 [R/W] B 00000000	—			
0000 0F80$_H$	BT8TMR[R]H 00000000 00000000		BT8TMCR[R/W] B,H -0000000 00000000		Base timer ch.8
0000 0F84$_H$	—	BT8STC[R/W]B 0000-000	—		
0000 0F88$_H$	BT8PCSR/BT8PRLL[R/W]H XXXXXXXX XXXXXXXX		BT8PDUT/BT8PRLH/BT8DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F8C$_H$	—				
0000 0F90$_H$	BT9TMR[R]H 00000000 00000000		BT9TMCR[R/W] B,H -0000000 00000000		Base timer ch.9
0000 0F94$_H$	—	BT9STC[R/W]B 0000-000	—		
0000 0F98$_H$	BT9PCSR/BT9PRLL[R/W]H XXXXXXXX XXXXXXXX		BT9PDUT/BT9PRLH/BT9DTBF[R/W]H XXXXXXXX XXXXXXXX		
0000 0F9C$_H$	—				

Table A-1 I/O map (20 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0FA0$_H$	BTATMR[R]H 00000000 00000000		BTATMCR[R/W] B,H -0000000 00000000		Base timer ch.10
0000 0FA4$_H$	—	BTASTC[R/W]B 0000-000	—		
0000 0FA8$_H$	BTAPCSR/BTAPRLL [R/W]H XXXXXXXX XXXXXXXX		BTAPDUT/BTAPRLH/BTADTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FAC$_H$	—				
0000 0FB0$_H$	BTBTMR[R]H 00000000 00000000		BTBTMCR[R/W] B,H -0000000 00000000		Base timer ch.11
0000 0FB4$_H$	—	BTBSTC[R/W]B 0000-000	—		
0000 0FB8$_H$	BTBPCSR/BTBPRLL [R/W]H XXXXXXXX XXXXXXXX		BTBPDUT/BTBPRLH/BTBDTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FBC$_H$	BTSEL89AB [R/W] B 00000000	—			
0000 0FC0$_H$	BTCTMR[R]H 00000000 00000000		BTCTMCR[R/W] B,H -0000000 00000000		Base timer ch.12
0000 0FC4$_H$	—	BTCSTC[R/W]B 0000-000	—		
0000 0FC8$_H$	BTCPCSR/BTCPRLL[R/W]H XXXXXXXX XXXXXXXX		BTCPDUT/BTCPRLH/BTCDTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FCC$_H$	—				
0000 0FD0$_H$	BTDTMR[R]H 00000000 00000000		BTDTMCR[R/W] B,H -0000000 00000000		Base timer ch.13
0000 0FD4$_H$	—	BTDSTC[R/W]B 0000-000	—		
0000 0FD8$_H$	BTDPCSR/BTDPRLL[R/W]H XXXXXXXX XXXXXXXX		BTDPDUT/BTDPRLH/BTDDTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FDC$_H$	—				

MB91660 Series

Table A-1 I/O map (21 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 0FE0$_H$	BTETMR[R]H 00000000 00000000		BTETMCR[R/W] B,H -0000000 00000000		Base timer ch.14
0000 0FE4$_H$	—	BTESTC[R/W]B 0000-000	—		
0000 0FE8$_H$	BTEPCSR/BTEPRLL[R/W]H XXXXXXXX XXXXXXXX		BTEPDUT/BTEPRLH/BTEDTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FEC$_H$	—				
0000 0FF0$_H$	BTFTMR[R]H 00000000 00000000		BTFTMCR[R/W] B,H -0000000 00000000		Base timer ch.15
0000 0FF4$_H$	—	BTFSTC[R/W]B 0000-000	—		
0000 0FF8$_H$	BTFPCSR/BTFPRLL[R/W]H XXXXXXXX XXXXXXXX		BTFPDUT/BTFPRLH/BTFDTBF [R/W]H XXXXXXXX XXXXXXXX		
0000 0FFC$_H$	BTSELCDEF [R/W] B 00000000	—	BTSSSR [W] H XXXXXXXX XXXXXXXX		
0000 1000$_H$ to 0000 20FC$_H$	—				Reserved
0000 2100$_H$	HCNT1[R/W] B,H - - - - - 001	HCNT0[R/W] B,H 00000000	—		USB function with Mini- HOST
0000 2104$_H$	HERR[R/W] B,H 00000011	HIRQ[R/W] B,H 0- 000000	—		
0000 2108$_H$	HFCOMP[R/W] B,H 00000000	HSTATE[R,R/W] B,H - - - 10010	—		
0000 210C$_H$	HRTIMER1[R/W] B,H 00000000	HRTIMER0[R/W] B,H 00000000	—		
0000 2110$_H$	HADR[R/W] B,H - 0000000	HRTIMER2[R/W] B,H - - - - - - 00	—		
0000 2114$_H$	HEOF1[R/W] B,H - - 000000	HEOF0[R/W] B,H 00000000	—		
0000 2118$_H$	HFRAME1[R/W] B,H - - - - - 000	HFRAME0[R/W] B,H 00000000	—		
0000 211C$_H$	—	HTOKEN[R/W] B 00000000	—		

Table A-1 I/O map (22 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 2120$_H$	—	UDCC[R/W] B 1010- - 00	—		USB function with Mini- HOST
0000 2124$_H$	EP0C[R/W] H - - - - - - 0- - 1000000		—		
0000 2128$_H$	EP1C[R/W] H 01100001 00000000		—		
0000 212C$_H$	EP2C[R/W] H 0110000- - 1000000		—		
0000 2130$_H$	EP3C[R/W] H 0110000- - 1000000		—		
0000 2134$_H$	EP4C[R/W] H 0110000- - 1000000		—		
0000 2138$_H$	EP5C[R/W] H 0110000- - 1000000		—		
0000 213C$_H$	TMSP[R] H - - - - - 000 00000000		—		
0000 2140$_H$	UDCIE[R,R/W] B,H - - 000000	UDCS[R/W] B,H - - 000000	—		
0000 2144$_H$	EP0IS[R/W] H 10- - - 1- - - - - - - - -		—		
0000 2148$_H$	EP00S[R,R/W] H 100- - 00- - XXXXXXX		—		
0000 214C$_H$	EP1S[R,R/W] H 100- 000X XXXXXXXX		—		
0000 2150$_H$	EP2S[R,R/W] H 100- 000- - XXXXXXX		—		
0000 2154$_H$	EP3S[R,R/W] H 100- 000- - XXXXXXX		—		
0000 2158$_H$	EP4S[R,R/W] H 100- 000- - XXXXXXX		—		
0000 215C$_H$	EP5S[R,R/W] H 100- 000- - XXXXXXX		—		
0000 2160$_H$	EP0DTH [R/W] B,H XXXXXXXX	EP0DTL [R/W] B,H XXXXXXXX	—		
0000 2164$_H$	EP1DTH [R/W] B,H XXXXXXXX	EP1DTL [R/W] B,H XXXXXXXX	—		
0000 2168$_H$	EP2DTH [R/W] B,H XXXXXXXX	EP2DTL [R/W] B,H XXXXXXXX	—		
0000 216C$_H$	EP3DTH [R/W] B,H XXXXXXXX	EP3DTL [R/W] B,H XXXXXXXX	—		
0000 2170$_H$	EP4DTH [R/W] B,H XXXXXXXX	EP4DTL [R/W] B,H XXXXXXXX	—		

MB91660 Series

Table A-1 I/O map (23 / 23)

Address	Registers				Block
	+0	+1	+2	+3	
0000 2174$_H$	EP5DTH [R/W] B,H XXXXXXXX	EP5DTL [R/W] B,H XXXXXXXX	—		USB function with Mini- HOST
0000 2178$_H$ to 0000 217C$_H$	—				
0000 2180$_H$	SLIFRXDR [R] W 00000000 00000000 00000000 00000000				Slave interface
0000 2184$_H$	SLIFTXDR [W] W 00000000 00000000 00000000 00000000				
0000 2188$_H$	SLIFRXR[R] W 00000000 00000000 - - - - - - - - - - - - - - - -				
0000 218C$_H$	SLIFTXR[W] W 00000000 00000000 - - - - - - - - - - - - - - - -				
0000 2190$_H$	SLIFCR[W] W - 0 00000000				
0000 2194$_H$	SLIFSR[R] W - 00000000				
0000 2198$_H$	SLIFRXSR[R] W 00000000 00000000 00000000 00000000				
0000 219C$_H$	SLIFTXSR[R] W 00000000 00000000 - - - - - - - - - - - - - - - -				
0000 21A0$_H$	SLIFDRXR[R] W 00000000 00000000 00000000 00000000				
0000 21A4$_H$	DREQSEL [R/W] B,H 00111011	USBSEL [R/W] B,H - - - - - - - 0	USBEN [R/W] B - - - - - - - 0	—	DMA transfer request selector/USB enable
0000 21A8$_H$ to 0000 FFFF$_H$	—				Reserved

*1 Byte access is available only when accessing the lower 8 bits within 9 bits.

*2 The register of I^2C can not be read immediate after reset.

*3 Value just after reset by $\overline{INIT}$ pin.

Do not access the reserved areas.

APPENDIX B List of Registers

This appendix lists the registers that can be used with this series.

The list is sorted in alphabetical order by abbreviated register name of this series.

Abbreviated Name	Register Name	Address	Reference
A			
ACR0	Area configuration register 0	0000 0640_H	13.4.2
ACR1	Area configuration register 1	0000 0644_H	13.4.2
ACR2	Area configuration register 2	0000 0648_H	13.4.2
ACR3	Area configuration register 3	0000 $064C_H$	13.4.2
ADCHE	A/D channel enable register	0000 $01AC_H$	15.4.6
ADCR0	A/DC control register 0	0000 0120_H	26.4.1
ADCT0	Compare time setting register 0	0000 $013A_H$	26.4.15
ADSR0	A/DC status register 0	0000 0121_H	26.4.2
ADSS00	Sampling time select register 00	0000 0137_H	26.4.14
ADSS10	Sampling time select register 10	0000 0136_H	26.4.14
ADSS20	Sampling time select register 20	0000 0135_H	26.4.14
ADST00	Sampling time setting register 00	0000 0138_H	26.4.13
ADST10	Sampling time setting register 10	0000 0139_H	26.4.13
ASR0	Area setting register 0	0000 0600_H	13.4.1
ASR1	Area setting register 1	0000 0604_H	13.4.1
ASR2	Area setting register 2	0000 0608_H	13.4.1
ASR3	Area setting register 3	0000 $060C_H$	13.4.1
AWR0	Area wait register 0	0000 0680_H	13.4.3
AWR1	Area wait register 1	0000 0684_H	13.4.3

AWR2	Area wait register 2	0000 0688$_H$	13.4.3
AWR3	Area wait register 3	0000 068C$_H$	13.4.3

B

BGR00	Baud rate generator register 00	0000 0067$_H$	28.4.6, 28.12.6
BGR01	Baud rate generator register 01	0000 006F$_H$	28.4.6, 28.12.6, 28.19.8
BGR02	Baud rate generator register 02	0000 007B$_H$	28.4.6, 28.12.6, 28.19.8
BGR03	Baud rate generator register 03	0000 0087$_H$	28.4.6, 28.12.6, 28.19.8
BGR04	Baud rate generator register 04	0000 0093$_H$	28.4.6, 28.12.6, 28.19.8
BGR05	Baud rate generator register 05	0000 009F$_H$	28.4.6, 28.12.6, 28.19.8
BGR06	Baud rate generator register 06	0000 00AB$_H$	28.4.6, 28.12.6, 28.19.8
BGR07	Baud rate generator register 07	0000 00B7$_H$	28.4.6, 28.12.6, 28.19.8
BGR08	Baud rate generator register 08	0000 00D7$_H$	28.4.6, 28.12.6, 28.19.8
BGR09	Baud rate generator register 09	0000 00E7$_H$	28.4.6, 28.12.6, 28.19.8
BGR010	Baud rate generator register 010	0000 00F7$_H$	28.4.6, 28.12.6, 28.19.8
BGR011	Baud rate generator register 011	0000 0107$_H$	28.4.6, 28.12.6, 28.19.8
BGR10	Baud rate generator register 10	0000 0066$_H$	28.4.6, 28.12.6
BGR11	Baud rate generator register 11	0000 006E$_H$	28.4.6, 28.12.6, 28.19.8
BGR12	Baud rate generator register 12	0000 007A$_H$	28.4.6, 28.12.6, 28.19.8
BGR13	Baud rate generator register 13	0000 0086$_H$	28.4.6, 28.12.6, 28.19.8
BGR14	Baud rate generator register 14	0000 0092$_H$	28.4.6, 28.12.6, 28.19.8
BGR15	Baud rate generator register 15	0000 009E$_H$	28.4.6, 28.12.6, 28.19.8
BGR16	Baud rate generator register 16	0000 00AA$_H$	28.4.6, 28.12.6, 28.19.8
BGR17	Baud rate generator register 17	0000 00B6$_H$	28.4.6, 28.12.6, 28.19.8
BGR18	Baud rate generator register 18	0000 00D6$_H$	28.4.6, 28.12.6, 28.19.8
BGR19	Baud rate generator register 19	0000 00E6$_H$	28.4.6, 28.12.6, 28.19.8
BGR110	Baud rate generator register 110	0000 00F6$_H$	28.4.6, 28.12.6, 28.19.8
BGR111	Baud rate generator register 111	0000 0106$_H$	28.4.6, 28.12.6, 28.19.8
BT0DTBF	Base timer 0 data buffer register	0000 014A$_H$	24.8.4.2

BT0PCSR	Base timer 0 cycle setting register	0000 0148$_H$	24.8.1.2, 24.8.3.2
BT0PDUT	Base timer 0 duty setting register	0000 014A$_H$	24.8.1.3
BT0PRLH	Base timer 0 H width setting register	0000 014A$_H$	24.8.2.3
BT0PRLL	Base timer 0 L width setting register	0000 0148$_H$	24.8.2.2
BT0STC	Base timer 0 status control register	0000 0145$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT0TMCR	Base timer 0 timer control register	0000 0142$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT0TMR	Base timer 0 timer register	0000 0140$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT1DTBF	Base timer 1 data buffer register	0000 015A$_H$	24.8.4.2
BT1PCSR	Base timer 1 cycle setting register	0000 0158$_H$	24.8.1.2, 24.8.3.2
BT1PDUT	Base timer 1 duty setting register	0000 015A$_H$	24.8.1.3
BT1PRLH	Base timer 1 H width setting register	0000 015A$_H$	24.8.2.3
BT1PRLL	Base timer 1 L width setting register	0000 0158$_H$	24.8.2.2
BT1STC	Base timer 1 status control register	0000 0155$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT1TMCR	Base timer 1 timer control register	0000 0152$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT1TMR	Base timer 1 timer register	0000 0150$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT2DTBF	Base timer 2 data buffer register	0000 016A$_H$	24.8.4.2
BT2PCSR	Base timer 2 cycle setting register	0000 0168$_H$	24.8.1.2, 24.8.3.2
BT2PDUT	Base timer 2 duty setting register	0000 016A$_H$	24.8.1.3
BT2PRLH	Base timer 2 H width setting register	0000 016A$_H$	24.8.2.3
BT2PRLL	Base timer 2 L width setting register	0000 0168$_H$	24.8.2.2
BT2STC	Base timer 2 status control register	0000 0165$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT2TMCR	Base timer 2 timer control register	0000 0162$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT2TMR	Base timer 2 timer register	0000 0160$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT3DTBF	Base timer 3 data buffer register	0000 017A$_H$	24.8.4.2
BT3PCSR	Base timer 3 cycle setting register	0000 0178$_H$	24.8.1.2, 24.8.3.2
BT3PDUT	Base timer 3 duty setting register	0000 017A$_H$	24.8.1.3
BT3PRLH	Base timer 3 H width setting register	0000 017A$_H$	24.8.2.3
BT3PRLL	Base timer 3 L width setting register	0000 0178$_H$	24.8.2.2
BT3STC	Base timer 3 status control register	0000 0175$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT3TMCR	Base timer 3 timer control register	0000 0172$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT3TMR	Base timer 3 timer register	0000 0170$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3

BT4DTBF	Base timer 4 data buffer register	0000 0F4A$_H$	24.8.4.2
BT4PCSR	Base timer 4 cycle setting register	0000 0F48$_H$	24.8.1.2, 24.8.3.2
BT4PDUT	Base timer 4 duty setting register	0000 0F4A$_H$	24.8.1.3
BT4PRLH	Base timer 4 H width setting register	0000 0F4A$_H$	24.8.2.3
BT4PRLL	Base timer 4 L width setting register	0000 0F48$_H$	24.8.2.2
BT4STC	Base timer 4 status control register	0000 0F45$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT4TMCR	Base timer 4 timer control register	0000 0F42$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT4TMR	Base timer 4 timer register	0000 0F40$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT5DTBF	Base timer 5 data buffer register	0000 0F5A$_H$	24.8.4.2
BT5PCSR	Base timer 5 cycle setting register	0000 0F58$_H$	24.8.1.2, 24.8.3.2
BT5PDUT	Base timer 5 duty setting register	0000 0F5A$_H$	24.8.1.3
BT5PRLH	Base timer 5 H width setting register	0000 0F5A$_H$	24.8.2.3
BT5PRLL	Base timer 5 L width setting register	0000 0F58$_H$	24.8.2.2
BT5STC	Base timer 5 status control register	0000 0F55$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT5TMCR	Base timer 5 timer control register	0000 0F52$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT5TMR	Base timer 5 timer register	0000 0F50$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT6DTBF	Base timer 6 data buffer register	0000 0F6A$_H$	24.8.4.2
BT6PCSR	Base timer 6 cycle setting register	0000 0F68$_H$	24.8.1.2, 24.8.3.2
BT6PDUT	Base timer 6 duty setting register	0000 0F6A$_H$	24.8.1.3
BT6PRLH	Base timer 6 H width setting register	0000 0F6A$_H$	24.8.2.3
BT6PRLL	Base timer 6 L width setting register	0000 0F68$_H$	24.8.2.2
BT6STC	Base timer 6 status control register	0000 0F65$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT6TMCR	Base timer 6 timer control register	0000 0F62$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT6TMR	Base timer 6 timer register	0000 0F60$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT7DTBF	Base timer 7 data buffer register	0000 0F7A$_H$	24.8.4.2
BT7PCSR	Base timer 7 cycle setting register	0000 0F78$_H$	24.8.1.2, 24.8.3.2
BT7PDUT	Base timer 7 duty setting register	0000 0F7A$_H$	24.8.1.3
BT7PRLH	Base timer 7 H width setting register	0000 0F7A$_H$	24.8.2.3
BT7PRLL	Base timer 7 L width setting register	0000 0F78$_H$	24.8.2.2
BT7STC	Base timer 7 status control register	0000 0F75$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT7TMCR	Base timer 7 timer control register	0000 0F72$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1

BT7TMR	Base timer 7 timer register	0000 0F70$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT8DTBF	Base timer 8 data buffer register	0000 0F8A$_H$	24.8.4.2
BT8PCSR	Base timer 8 cycle setting register	0000 0F88$_H$	24.8.1.2, 24.8.3.2
BT8PDUT	Base timer 8 duty setting register	0000 0F8A$_H$	24.8.1.3
BT8PRLH	Base timer 8 H width setting register	0000 0F8A$_H$	24.8.2.3
BT8PRLL	Base timer 8 L width setting register	0000 0F88$_H$	24.8.2.2
BT8STC	Base timer 8 status control register	0000 0F85$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT8TMCR	Base timer 8 timer control register	0000 0F82$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT8TMR	Base timer 8 timer register	0000 0F80$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BT9DTBF	Base timer 9 data buffer register	0000 0F9A$_H$	24.8.4.2
BT9PCSR	Base timer 9 cycle setting register	0000 0F98$_H$	24.8.1.2, 24.8.3.2
BT9PDUT	Base timer 9 duty setting register	0000 0F9A$_H$	24.8.1.3
BT9PRLH	Base timer 9 H width setting register	0000 0F9A$_H$	24.8.2.3
BT9PRLL	Base timer 9 L width setting register	0000 0F98$_H$	24.8.2.2
BT9STC	Base timer 9 status control register	0000 0F95$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT9TMCR	Base timer 9 timer control register	0000 0F92$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BT9TMR	Base timer 9 timer register	0000 0F90$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTADTBF	Base timer A data buffer register	0000 0FAA$_H$	24.8.4.2
BTAPCSR	Base timer A cycle setting register	0000 0FA8$_H$	24.8.1.2, 24.8.3.2
BTAPDUT	Base timer A duty setting register	0000 0FAA$_H$	24.8.1.3
BTAPRLH	Base timer A H width setting register	0000 0FAA$_H$	24.8.2.3
BTAPRLL	Base timer A L width setting register	0000 0FA8$_H$	24.8.2.2
BTASTC	Base timer A status control register	0000 0FA5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTATMCR	Base timer A timer control register	0000 0FA2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTATMR	Base timer A timer register	0000 0FA0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTBDTBF	Base timer B data buffer register	0000 0FBA$_H$	24.8.4.2
BTBPCSR	Base timer B cycle setting register	0000 0FB8$_H$	24.8.1.2, 24.8.3.2
BTBPDUT	Base timer B duty setting register	0000 0FBA$_H$	24.8.1.3
BTBPRLH	Base timer B H width setting register	0000 0FBA$_H$	24.8.2.3
BTBPRLL	Base timer B L width setting register	0000 0FB8$_H$	24.8.2.2
BTBSTC	Base timer B status control register	0000 0FB5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1

BTBTMCR	Base timer B timer control register	0000 0FB2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTBTMR	Base timer B timer register	0000 0FB0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTCDTBF	Base timer C data buffer register	0000 0FCA$_H$	24.8.4.2
BTCPCSR	Base timer C cycle setting register	0000 0FC8$_H$	24.8.1.2, 24.8.3.2
BTCPDUT	Base timer C duty setting register	0000 0FCA$_H$	24.8.1.3
BTCPRLH	Base timer C H width setting register	0000 0FCA$_H$	24.8.2.3
BTCPRLL	Base timer C L width setting register	0000 0FC8$_H$	24.8.2.2
BTCSTC	Base timer C status control register	0000 0FC5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTCTMCR	Base timer C timer control register	0000 0FC2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTCTMR	Base timer C timer register	0000 0FC0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTDDTBF	Base timer D data buffer register	0000 0FDA$_H$	24.8.4.2
BTDPCSR	Base timer D cycle setting register	0000 0FD8$_H$	24.8.1.2, 24.8.3.2
BTDPDUT	Base timer D duty setting register	0000 0FDA$_H$	24.8.1.3
BTDPRLH	Base timer D H width setting register	0000 0FDA$_H$	24.8.2.3
BTDPRLL	Base timer D L width setting register	0000 0FD8$_H$	24.8.2.2
BTDSTC	Base timer D status control register	0000 0FD5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTDTMCR	Base timer D timer control register	0000 0FD2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTDTMR	Base timer D timer register	0000 0FD0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTEDTBF	Base timer E data buffer register	0000 0FEA$_H$	24.8.4.2
BTEPCSR	Base timer E cycle setting register	0000 0FE8$_H$	24.8.1.2, 24.8.3.2
BTEPDUT	Base timer E duty setting register	0000 0FEA$_H$	24.8.1.3
BTEPRLH	Base timer E H width setting register	0000 0FEA$_H$	24.8.2.3
BTEPRLL	Base timer E L width setting register	0000 0FE8$_H$	24.8.2.2
BTESTC	Base timer E status control register	0000 0FE5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTETMCR	Base timer E timer control register	0000 0FE2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTETMR	Base timer E timer register	0000 0FE0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTFDTBF	Base timer F data buffer register	0000 0FFA$_H$	24.8.4.2
BTFPCSR	Base timer F cycle setting register	0000 0FF8$_H$	24.8.1.2, 24.8.3.2
BTFPDUT	Base timer F duty setting register	0000 0FFA$_H$	24.8.1.3
BTFPRLH	Base timer F H width setting register	0000 0FFA$_H$	24.8.2.3
BTFPRLL	Base timer F L width setting register	0000 0FF8$_H$	24.8.2.2

BTFSTC	Base timer F status control register	0000 0FF5$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTFTMCR	Base timer F timer control register	0000 0FF2$_H$	24.8.1.1, 24.8.2.1, 24.8.3.1, 24.8.4.1
BTFTMR	Base timer F timer register	0000 0FF0$_H$	24.8.1.4, 24.8.2.4, 24.8.3.3
BTSEL0123	Base timer io select register for ch.0/1/2/3	0000 017C$_H$	23.4.1
BTSEL4567	Base timer io select register for ch.4/5/6/7	0000 0F7C$_H$	23.4.2
BTSEL89AB	Base timer io select register for ch.8/9/A/B	0000 0FBC$_H$	23.4.3
BTSELCDEF	Base timer io select register for ch.C/D/E/F	0000 0FFC$_H$	23.4.4
BTSSSR	Base timer same time soft start register	0000 0FFE$_H$	23.4.5

C

CCR0	Counter control register 0	0000 01C4$_H$	26.4.12
CCR1	Counter control register 1	0000 01D4$_H$	26.4.12
CCR2	Counter control register 2	0000 01E4$_H$	26.4.12
CCR3	Counter control register 3	0000 01F4$_H$	26.4.12
CMONR	Clock source monitor register	0000 0511$_H$	4.4.2
CMPCR0	A/D comparison control register 0	0000 0133$_H$	26.4.12
CMPD0	A/D comparison data setting register 0	0000 0132$_H$	26.4.11
CPCLR0	Compare clear register 0	0000 0200$_H$	19.4.2
CPCLR1	Compare clear register 1	0000 0260$_H$	19.4.2
CSELR	Clock source select register	0000 0510$_H$	4.4.1
CSR0	Counter status register 0	0000 01C7$_H$	25.4.4
CSR1	Counter status register 1	0000 01D7$_H$	25.4.4
CSR2	Counter status register 2	0000 01E7$_H$	25.4.4
CSR3	Counter status register 3	0000 01F7$_H$	25.4.4
CSTBR	Clock stabilization time select register	0000 0516$_H$	4.4.3

D

DACR0	D/A control register 0	0000 0180$_H$	27.4.2
DACR1	D/A control register 1	0000 0182$_H$	27.4.2

DACR2	D/A control register 2	0000 0184_H	27.4.2
DADR0	D/A data register 0	0000 0181_H	27.4.1
DADR1	D/A data register 1	0000 0183_H	27.4.1
DADR2	D/A data register 2	0000 0185_H	27.4.1
DCCR0	DMA channel control register 0	0000 $0C00_H$	33.4.5
DCCR1	DMA channel control register 1	0000 $0C10_H$	33.4.5
DCCR2	DMA channel control register 2	0000 $0C20_H$	33.4.5
DCCR3	DMA channel control register 3	0000 $0C30_H$	33.4.5
DCCR4	DMA channel control register 4	0000 $0C40_H$	33.4.5
DCCR5	DMA channel control register 5	0000 $0C50_H$	33.4.5
DCCR6	DMA channel control register 6	0000 $0C60_H$	33.4.5
DCCR7	DMA channel control register 7	0000 $0C70_H$	33.4.5
DCSR0	DMA channel status register 0	0000 $0C04_H$	33.4.6
DCSR1	DMA channel status register 1	0000 $0C14_H$	33.4.6
DCSR2	DMA channel status register 2	0000 $0C24_H$	33.4.6
DCSR3	DMA channel status register 3	0000 $0C34_H$	33.4.6
DCSR4	DMA channel status register 4	0000 $0C44_H$	33.4.6
DCSR5	DMA channel status register 5	0000 $0C54_H$	33.4.6
DCSR6	DMA channel status register 6	0000 $0C64_H$	33.4.6
DCSR7	DMA channel status register 7	0000 $0C74_H$	33.4.6
DDAR0	DMA destination address register 0	0000 $0C0C_H$	33.4.3
DDAR1	DMA destination address register 1	0000 $0C1C_H$	33.4.3
DDAR2	DMA destination address register 2	0000 $0C2C_H$	33.4.3
DDAR3	DMA destination address register 3	0000 $0C3C_H$	33.4.3
DDAR4	DMA destination address register 4	0000 $0C4C_H$	33.4.3
DDAR5	DMA destination address register 5	0000 $0C5C_H$	33.4.3
DDAR6	DMA destination address register 6	0000 $0C6C_H$	33.4.3
DDAR7	DMA destination address register 7	0000 $0C7C_H$	33.4.3
DDR0	Port data direction register 0	0000 0400_H	15.4.1
DDR1	Port data direction register 1	0000 0401_H	15.4.1
DDR2	Port data direction register 2	0000 0402_H	15.4.1

DDR3	Port data direction register 3	0000 0403_H	15.4.1
DDR4	Port data direction register 4	0000 0404_H	15.4.1
DDR5	Port data direction register 5	0000 0405_H	15.4.1
DDR6	Port data direction register 6	0000 0406_H	15.4.1
DDR7	Port data direction register 7	0000 0407_H	15.4.1
DDR8	Port data direction register 8	0000 0408_H	15.4.1
DDR9	Port data direction register 9	0000 0409_H	15.4.1
DDRA	Port data direction register A	0000 $040A_H$	15.4.1
DDRG	Port data direction register G	0000 0410_H	15.4.1
DDRH	Port data direction register H	0000 0411_H	15.4.1
DDRK	Port data direction register K	0000 0414_H	15.4.1
DICR	Delayed interrupt control register	0000 0044_H	12.3.1
DILVR	DMA-halt by interrupt level register	0000 $0DF7_H$	33.4.7
DIVR0	Divide clock configuration register 0	0000 0488_H	5.4.1
DIVR1	Divide clock configuration register 1	0000 0489_H	5.4.2
DIVR2	Divide clock configuration register 2	0000 $048A_H$	5.4.3
DMACR	DMA control register	0000 $0DF8_H$	33.4.1
DMAR0	DMA transfer register 0	0000 $06C0_H$	13.4.4
DMAR1	DMA transfer register 1	0000 $06C4_H$	13.4.4
DMAR2	DMA transfer register 2	0000 $06C8_H$	13.4.4
DMAR3	DMA transfer register 3	0000 $06CC_H$	13.4.4
DREQSEL	DREQ select register	0000 $21A4_H$	30.3.1
DSAR0	DMA source address register 0	0000 $0C08_H$	33.4.2
DSAR1	DMA source address register 1	0000 $0C18_H$	33.4.2
DSAR2	DMA source address register 2	0000 $0C28_H$	33.4.2
DSAR3	DMA source address register 3	0000 $0C38_H$	33.4.2
DSAR4	DMA source address register 4	0000 $0C48_H$	33.4.2
DSAR5	DMA source address register 5	0000 $0C58_H$	33.4.2
DSAR6	DMA source address register 6	0000 $0C68_H$	33.4.2
DSAR7	DMA source address register 7	0000 $0C78_H$	33.4.2
DTCR0	DMA transfer count register 0	0000 $0C06_H$	33.4.4

DTCR1	DMA transfer count register 1	0000 $0C16_H$	33.4.4
DTCR2	DMA transfer count register 2	0000 $0C26_H$	33.4.4
DTCR3	DMA transfer count register 3	0000 $0C36_H$	33.4.4
DTCR4	DMA transfer count register 4	0000 $0C46_H$	33.4.4
DTCR5	DMA transfer count register 5	0000 $0C56_H$	33.4.4
DTCR6	DMA transfer count register 6	0000 $0C66_H$	33.4.4
DTCR7	DMA transfer count register 7	0000 $0C76_H$	33.4.4

E

EIRR0	External interrupt request register 0	0000 0040_H	16.4.2
EIRR1	External interrupt request register 1	0000 0110_H	16.4.2
EIRR2	External interrupt request register 2	0000 0114_H	16.4.2
EIRR3	External interrupt request register 3	0000 0118_H	16.4.2
ELVR0	External interrupt request level register 0	0000 0042_H	16.4.1
ELVR1	External interrupt request level register 1	0000 0112_H	16.4.1
ELVR2	External interrupt request level register 2	0000 0116_H	16.4.1
ELVR3	External interrupt request level register 3	0000 $011A_H$	16.4.1
ENIR0	Enable interrupt request register 0	0000 0041_H	16.4.3
ENIR1	Enable interrupt request register 1	0000 0111_H	16.4.3
ENIR2	Enable interrupt request register 2	0000 0115_H	16.4.3
ENIR3	Enable interrupt request register 3	0000 0119_H	16.4.3
EP0C	EP0 control register	0000 2124_H	31.3.4
EP0DTH	EP0 data register upper	0000 2160_H	31.3.12
EP0DTL	EP0 data register lower	0000 2161_H	31.3.12
EP0IS	EP0I status register	0000 2144_H	31.3.9
EP0OS	EP0O status register	0000 2148_H	31.3.10
EP1C	EP1 control register	0000 2128_H	31.3.5
EP1DTH	EP1 data register upper	0000 2164_H	31.3.12
EP1DTL	EP1 data register lower	0000 2165_H	31.3.12
EP1S	EP1 status register	0000 $214C_H$	31.3.11

EP2C	EP2 control register	0000 212C$_H$	31.3.5
EP2DTH	EP2 data register upper	0000 2168$_H$	31.3.12
EP2DTL	EP2 data register lower	0000 2169$_H$	31.3.12
EP2S	EP2 status register	0000 2150$_H$	31.3.11
EP3C	EP3 control register	0000 2130$_H$	31.3.5
EP3DTH	EP3 data register upper	0000 216C$_H$	31.3.12
EP3DTL	EP3 data register lower	0000 216D$_H$	31.3.12
EP3S	EP3 status register	0000 2154$_H$	31.3.11
EP4C	EP4 control register	0000 2134$_H$	31.3.5
EP4DTH	EP4 data register upper	0000 2170$_H$	31.3.12
EP4DTL	EP4 data register lower	0000 2171$_H$	31.3.12
EP4S	EP4 status register	0000 2158$_H$	31.3.11
EP5C	EP5 control register	0000 2138$_H$	31.3.5
EP5DTH	EP5 data register upper	0000 2174$_H$	31.3.12
EP5DTL	EP5 data register lower	0000 2175$_H$	31.3.12
EP5S	EP5 status register	0000 215C$_H$	31.3.11
EPFR0	Extended port function register 0	0000 04B8$_H$	15.4.3
EPFR1	Extended port function register 1	0000 04B9$_H$	15.4.3
EPFR2	Extended port function register 2	0000 04BA$_H$	15.4.3
EPFR3	Extended port function register 3	0000 04BB$_H$	15.4.3
EPFR4	Extended port function register 4	0000 04BC$_H$	15.4.3
EPFR5	Extended port function register 5	0000 04BD$_H$	15.4.3
EPFR6	Extended port function register 6	0000 04BE$_H$	15.4.3
EPFR7	Extended port function register 7	0000 04BF$_H$	15.4.3
EPFR8	Extended port function register 8	0000 04C0$_H$	15.4.3
EPFR9	Extended port function register 9	0000 04C1$_H$	15.4.3
EPFR10	Extended port function register 10	0000 04C2$_H$	15.4.3
EPFR11	Extended port function register 11	0000 04C3$_H$	15.4.3
EPFR12	Extended port function register 12	0000 04C4$_H$	15.4.3
EPFR13	Extended port function register 13	0000 04C5$_H$	15.4.3
EPFR14	Extended port function register 14	0000 04C6$_H$	15.4.3

EPFR15	Extended port function register 15	0000 $04C7_H$	15.4.3
EPFR16	Extended port function register 16	0000 $04C8_H$	15.4.3
EPFR17	Extended port function register 17	0000 $04C9_H$	15.4.3
EPFR18	Extended port function register 18	0000 $04CA_H$	15.4.3
EPFR19	Extended port function register 19	0000 $04CB_H$	15.4.3
EPFR20	Extended port function register 20	0000 $04CC_H$	15.4.3
EPFR21	Extended port function register 21	0000 $04CD_H$	15.4.3
EPFR22	Extended port function register 22	0000 $04CE_H$	15.4.3
EPFR23	Extended port function register 23	0000 $04CF_H$	15.4.3
EPFR24	Extended port function register 24	0000 $04D0_H$	15.4.3
EPFR25	Extended port function register 25	0000 $04D1_H$	15.4.3
EPFR26	Extended port function register 26	0000 $04D2_H$	15.4.3
EPFR27	Extended port function register 27	0000 $04D3_H$	15.4.3
EPFR28	Extended port function register 28	0000 $04D4_H$	15.4.3
EPFR29	Extended port function register 29	0000 $04D5_H$	15.4.3
EPFR30	Extended port function register 30	0000 $04D6_H$	15.4.3
EPFR31	Extended port function register 31	0000 $04D7_H$	15.4.3
EPFR32	Extended port function register 32	0000 $04D8_H$	15.4.3
EPFR33	Extended port function register 33	0000 $04D9_H$	15.4.3
EPFR34	Extended port function register 34	0000 $04DA_H$	15.4.3
EPFR35	Extended port function register 35	0000 $04DB_H$	15.4.3
ESCR0	Extended serial control register 0	0000 0063_H	28.4.4, 28.12.4
ESCR1	Extended serial control register 1	0000 $006B_H$	28.4.4, 28.12.4
ESCR2	Extended serial control register 2	0000 0077_H	28.4.4, 28.12.4
ESCR3	Extended serial control register 3	0000 0083_H	28.4.4, 28.12.4
ESCR4	Extended serial control register 4	0000 $008F_H$	28.4.4, 28.12.4
ESCR5	Extended serial control register 5	0000 $009B_H$	28.4.4, 28.12.4
ESCR6	Extended serial control register 6	0000 $00A7_H$	28.4.4, 28.12.4
ESCR7	Extended serial control register 7	0000 $00B3_H$	28.4.4, 28.12.4
ESCR8	Extended serial control register 8	0000 $00D3_H$	28.4.4, 28.12.4
ESCR9	Extended serial control register 9	0000 $00E3_H$	28.4.4, 28.12.4

ESCR10	Extended serial control register 10	0000 00F3$_H$	28.4.4, 28.12.4
ESCR11	Extended serial control register 11	0000 0103$_H$	28.4.4, 28.12.4

F

FBYTE18	FIFO byte register 18	0000 00DF$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE19	FIFO byte register 19	0000 00EF$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE110	FIFO byte register 110	0000 00FF$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE111	FIFO byte register 111	0000 010F$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE28	FIFO byte register 28	0000 00DE$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE29	FIFO byte register 29	0000 00EE$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE210	FIFO byte register 210	0000 00FE$_H$	28.4.9, 28.12.9, 28.19.11
FBYTE211	FIFO byte register 211	0000 010E$_H$	28.4.9, 28.12.9, 28.19.11
FCR08	FIFO control register 08	0000 00DD$_H$	28.4.8, 28.12.8, 28.19.10
FCR09	FIFO control register 09	0000 00ED$_H$	28.4.8, 28.12.8, 28.19.10
FCR010	FIFO control register 010	0000 00FD$_H$	28.4.8, 28.12.8, 28.19.10
FCR011	FIFO control register 011	0000 010D$_H$	28.4.8, 28.12.8, 28.19.10
FCR18	FIFO control register 18	0000 00DC$_H$	28.4.7, 28.12.7, 28.19.9
FCR19	FIFO control register 19	0000 00EC$_H$	28.4.7, 28.12.7, 28.19.9
FCR110	FIFO control register 110	0000 00FC$_H$	28.4.7, 28.12.7, 28.19.9
FCR111	FIFO control register 111	0000 010C$_H$	28.4.7, 28.12.7, 28.19.9
FCTLR	FLASH control register	0000 0320$_H$	35.2.1, 36.3.2
FRTSEL	Free-run timer select register	0000 025C$_H$	19.4.1
FSTR	FLASH status register	0000 0323$_H$	36.3.1

H

HADR	Host address register	0000 2110$_H$	32.3.7
HCNT0	Host control register 0	0000 2101$_H$	32.3.1
HCNT1	Host control register 1	0000 2100$_H$	32.3.1
HEOF0	EOF setting register 0	0000 2115$_H$	32.3.8

HEOF1	EOF setting register 1	0000 2114_H	32.3.8
HERR	Host error status register	0000 2104_H	32.3.3
HFCOMP	SOF interrupt FRAME comparison register	0000 2108_H	32.3.5
HFRAME0	FRAME setting register 0	0000 2119_H	32.3.9
HFRAME1	FRAME setting register 1	0000 2118_H	32.3.9
HIRQ	Host interrupt register	0000 2105_H	32.3.2
HRTIMER0	Retry timer setting register 0	0000 $210D_H$	32.3.6
HRTIMER1	Retry timer setting register 1	0000 $210C_H$	32.3.6
HRTIMER2	Retry timer setting register 2	0000 2111_H	32.3.6
HSTATE	Host status register	0000 2109_H	32.3.4
HTOKEN	Host token end point register	0000 $211D_H$	32.3.10

I

IBCR1	I^2C bus control register 1	0000 0068_H	28.19.1
IBCR2	I^2C bus control register 2	0000 0074_H	28.19.1
IBCR3	I^2C bus control register 3	0000 0080_H	28.19.1
IBCR4	I^2C bus control register 4	0000 $008C_H$	28.19.1
IBCR5	I^2C bus control register 5	0000 0098_H	28.19.1
IBCR6	I^2C bus control register 6	0000 $00A4_H$	28.19.1
IBCR7	I^2C bus control register 7	0000 $00B0_H$	28.19.1
IBCR8	I^2C bus control register 8	0000 $00D0_H$	28.19.1
IBCR9	I^2C bus control register 9	0000 $00E0_H$	28.19.1
IBCR10	I^2C bus control register 10	0000 $00F0_H$	28.19.1
IBCR11	I^2C bus control register 11	0000 0100_H	28.19.1
IBSR1	I^2C bus status register 1	0000 $006B_H$	28.19.3
IBSR2	I^2C bus status register 2	0000 0077_H	28.19.3
IBSR3	I^2C bus status register 3	0000 0083_H	28.19.3
IBSR4	I^2C bus status register 4	0000 $008F_H$	28.19.3
IBSR5	I^2C bus status register 5	0000 $009B_H$	28.19.3

IBSR6	I^2C bus status register 6	0000 $00A7_H$	28.19.3
IBSR7	I^2C bus status register 7	0000 $00B3_H$	28.19.3
IBSR8	I^2C bus status register 8	0000 $00D3_H$	28.19.3
IBSR9	I^2C bus status register 9	0000 $00E3_H$	28.19.3
IBSR10	I^2C bus status register 10	0000 $00F3_H$	28.19.3
IBSR11	I^2C bus status register 11	0000 0103_H	28.19.3
ICR00	Interrupt control register 00	0000 0440_H	10.3.1
ICR01	Interrupt control register 01	0000 0441_H	10.3.1
ICR02	Interrupt control register 02	0000 0442_H	10.3.1
ICR03	Interrupt control register 03	0000 0443_H	10.3.1
ICR04	Interrupt control register 04	0000 0444_H	10.3.1
ICR05	Interrupt control register 05	0000 0445_H	10.3.1
ICR06	Interrupt control register 06	0000 0446_H	10.3.1
ICR07	Interrupt control register 07	0000 0447_H	10.3.1
ICR08	Interrupt control register 08	0000 0448_H	10.3.1
ICR09	Interrupt control register 09	0000 0449_H	10.3.1
ICR10	Interrupt control register 10	0000 $044A_H$	10.3.1
ICR11	Interrupt control register 11	0000 $044B_H$	10.3.1
ICR12	Interrupt control register 12	0000 $044C_H$	10.3.1
ICR13	Interrupt control register 13	0000 $044D_H$	10.3.1
ICR14	Interrupt control register 14	0000 $044E_H$	10.3.1
ICR15	Interrupt control register 15	0000 $044F_H$	10.3.1
ICR16	Interrupt control register 16	0000 0450_H	10.3.1
ICR17	Interrupt control register 17	0000 0451_H	10.3.1
ICR18	Interrupt control register 18	0000 0452_H	10.3.1
ICR19	Interrupt control register 19	0000 0453_H	10.3.1
ICR20	Interrupt control register 20	0000 0454_H	10.3.1
ICR21	Interrupt control register 21	0000 0455_H	10.3.1
ICR22	Interrupt control register 22	0000 0456_H	10.3.1
ICR23	Interrupt control register 23	0000 0457_H	10.3.1

ICR24	Interrupt control register 24	0000 0458_H	10.3.1
ICR25	Interrupt control register 25	0000 0459_H	10.3.1
ICR26	Interrupt control register 26	0000 $045A_H$	10.3.1
ICR27	Interrupt control register 27	0000 $045B_H$	10.3.1
ICR28	Interrupt control register 28	0000 $045C_H$	10.3.1
ICR29	Interrupt control register 29	0000 $045D_H$	10.3.1
ICR30	Interrupt control register 30	0000 $045E_H$	10.3.1
ICR31	Interrupt control register 31	0000 $045F_H$	10.3.1
ICR32	Interrupt control register 32	0000 0460_H	10.3.1
ICR33	Interrupt control register 33	0000 0461_H	10.3.1
ICR34	Interrupt control register 34	0000 0462_H	10.3.1
ICR35	Interrupt control register 35	0000 0463_H	10.3.1
ICR36	Interrupt control register 36	0000 0464_H	10.3.1
ICR37	Interrupt control register 37	0000 0465_H	10.3.1
ICR38	Interrupt control register 38	0000 0466_H	10.3.1
ICR39	Interrupt control register 39	0000 0467_H	10.3.1
ICR40	Interrupt control register 40	0000 0468_H	10.3.1
ICR41	Interrupt control register 41	0000 0469_H	10.3.1
ICR42	Interrupt control register 42	0000 $046A_H$	10.3.1
ICR43	Interrupt control register 43	0000 $046B_H$	10.3.1
ICR44	Interrupt control register 44	0000 $046C_H$	10.3.1
ICR45	Interrupt control register 45	0000 $046D_H$	10.3.1
ICR46	Interrupt control register 46	0000 $046E_H$	10.3.1
ICR47	Interrupt control register 47	0000 $046F_H$	10.3.1
ICS01	Input capture status control register 01	0000 $021D_H$	20.4.1
ICS23	Input capture status control register 23	0000 $021F_H$	20.4.1
ICS45	Input capture status control register 45	0000 0231_H	20.4.1
ICS67	Input capture status control register 67	0000 0233_H	20.4.1
ICSEL0	Select Register 0 for DMA Transfer Request Clear by a Peripheral function	0000 $04F0_H$	34.3.2
ICSEL1	Select Register 1 for DMA Transfer Request Clear by a Peripheral function	0000 $04F1_H$	34.3.3

ICSEL2	Select Register 2 for DMA Transfer Request Clear by a Peripheral function	0000 04F2$_H$	34.3.4
ICSEL3	Select Register 3 for DMA Transfer Request Clear by a Peripheral function	0000 04F3$_H$	34.3.5
ICSEL4	Select Register 4 for DMA Transfer Request Clear by a Peripheral function	0000 04F4$_H$	34.3.6
ICSEL5	Select Register 5 for DMA Transfer Request Clear by a Peripheral function	0000 04F5$_H$	34.3.7
ICSEL6	Select Register 6 for DMA Transfer Request Clear by a Peripheral function	0000 04F6$_H$	34.3.8
ICSEL7	Select Register 7 for DMA Transfer Request Clear by a Peripheral function	0000 04F7$_H$	34.3.9
ICSEL8	Select Register 8 for DMA Transfer Request Clear by a Peripheral function	0000 04F8$_H$	34.3.10
ICSEL9	Select Register 9 for DMA Transfer Request Clear by a Peripheral function	0000 04F9$_H$	34.3.11
ICSEL10	Select Register 10 for DMA Transfer Request Clear by a Peripheral function	0000 04FA$_H$	34.3.12
ICSEL11	Select Register 11 for DMA Transfer Request Clear by a Peripheral function	0000 04FB$_H$	34.3.13
ICSEL12	Select Register 12 for DMA Transfer Request Clear by a Peripheral function	0000 04FC$_H$	34.3.14
ICSEL13	Select Register 13 for DMA Transfer Request Clear by a Peripheral function	0000 04FD$_H$	34.3.15
ICSEL14	Select Register 14 for DMA Transfer Request Clear by a Peripheral function	0000 04FE$_H$	34.3.16
IORR0	IO-data request register 0	0000 0490$_H$	34.3.1
IORR1	IO-data request register 1	0000 0491$_H$	34.3.1
IORR2	IO-data request register 2	0000 0492$_H$	34.3.1
IORR3	IO-data request register 3	0000 0493$_H$	34.3.1
IORR4	IO-data request register 4	0000 0494$_H$	34.3.1
IORR5	IO-data request register 5	0000 0495$_H$	34.3.1
IORR6	IO-data request register 6	0000 0496$_H$	34.3.1
IORR7	IO-data request register 7	0000 0497$_H$	34.3.1
IPCP0	Input capture data register 0	0000 020C$_H$	20.4.2
IPCP1	Input capture data register 1	0000 0210$_H$	20.4.2
IPCP2	Input capture data register 2	0000 0214$_H$	20.4.2
IPCP3	Input capture data register 3	0000 0218$_H$	20.4.2

IPCP4	Input capture data register 4	0000 0220_H	20.4.2
IPCP5	Input capture data register 5	0000 0224_H	20.4.2
IPCP6	Input capture data register 6	0000 0228_H	20.4.2
IPCP7	Input capture data register 7	0000 $022C_H$	20.4.2
IRPR0H	Interrupt request batch-read register 0 upper	0000 $01B0_H$	11.3.1
IRPR1H	Interrupt request batch-read register 1 upper	0000 $01B2_H$	11.3.2
IRPR2H	Interrupt request batch-read register 2 upper	0000 $01B4_H$	11.3.3
IRPR3H	Interrupt request batch-read register 3 upper	0000 $01B6_H$	11.3.5
IRPR4H	Interrupt request batch-read register 4 upper	0000 $01B8_H$	11.3.7
IRPR5H	Interrupt request batch-read register 5 upper	0000 $01BA_H$	11.3.9
IRPR6H	Interrupt request batch-read register 6 upper	0000 $01BC_H$	11.3.11
IRPR7H	Interrupt request batch-read register 7 upper	0000 $01BE_H$	11.3.13
IRPR1L	Interrupt request batch-read register 1 lower	0000 $01B3_H$	11.3.2
IRPR2L	Interrupt request batch-read register 2 lower	0000 $01B5_H$	11.3.4
IRPR3L	Interrupt request batch-read register 3 lower	0000 $01B7_H$	11.3.6
IRPR4L	Interrupt request batch-read register 4 lower	0000 $01B9_H$	11.3.8
IRPR5L	Interrupt request batch-read register 5 lower	0000 $01BB_H$	11.3.10
IRPR6L	Interrupt request batch-read register 6 lower	0000 $01BD_H$	11.3.12
IRPR7L	Interrupt request batch-read register 7 lower	0000 $01BF_H$	11.3.14
ISBA1	7 bit slave address register 1	0000 0071_H	28.19.7
ISBA2	7 bit slave address register 2	0000 $007D_H$	28.19.7
ISBA3	7 bit slave address register 3	0000 0089_H	28.19.7
ISBA4	7 bit slave address register 4	0000 0095_H	28.19.7
ISBA5	7 bit slave address register 5	0000 $00A1_H$	28.19.7
ISBA6	7 bit slave address register 6	0000 $00AD_H$	28.19.7
ISBA7	7 bit slave address register 7	0000 $00B9_H$	28.19.7
ISBA8	7-bit slave address register 8	0000 $00D9_H$	28.19.7
ISBA9	7-bit slave address register 9	0000 $00E9_H$	28.19.7
ISBA10	7-bit slave address register 10	0000 $00F9_H$	28.19.7
ISBA11	7-bit slave address register 11	0000 0109_H	28.19.7
ISMK1	7-bit slave address mask register 1	0000 0070_H	28.19.6

ISMK2	7-bit slave address mask register 2	0000 007C$_H$	28.19.6
ISMK3	7-bit slave address mask register 3	0000 0088$_H$	28.19.6
ISMK4	7-bit slave address mask register 4	0000 0094$_H$	28.19.6
ISMK5	7-bit slave address mask register 5	0000 00A0$_H$	28.19.6
ISMK6	7-bit slave address mask register 6	0000 00AC$_H$	28.19.6
ISMK7	7-bit slave address mask register 7	0000 00B8$_H$	28.19.6
ISMK8	7-bit slave address mask register 8	0000 00D8$_H$	28.19.6
ISMK9	7-bit slave address mask register 9	0000 00E8$_H$	28.19.6
ISMK10	7-bit slave address mask register 10	0000 00F8$_H$	28.19.6
ISMK11	7-bit slave address mask register 11	0000 0108$_H$	28.19.6

M

MTMCR	Main timer control register	0000 0512$_H$	6.3.1

O

OCCP0	Output compare register 0	0000 0234$_H$	21.4.1
OCCP1	Output compare register 1	0000 0238$_H$	21.4.1
OCCP2	Output compare register 2	0000 023C$_H$	21.4.1
OCCP3	Output compare register 3	0000 0240$_H$	21.4.1
OCCP4	Output compare register 4	0000 0248$_H$	21.4.1
OCCP5	Output compare register 5	0000 024C$_H$	21.4.1
OCCP6	Output compare register 6	0000 0250$_H$	21.4.1
OCCP7	Output compare register 7	0000 0254$_H$	21.4.1
OCSH1	Compare control register upper 1	0000 0244$_H$	21.4.2
OCSH3	Compare control register upper 3	0000 0246$_H$	21.4.2
OCSH5	Compare control register upper 5	0000 0258$_H$	21.4.2
OCSH7	Compare control register upper 7	0000 025A$_H$	21.4.2
OCSL0	Compare control register lower 0	0000 0245$_H$	21.4.3
OCSL2	Compare control register lower 2	0000 0247$_H$	21.4.3

OCSL4	Compare control register lower 4	0000 0259$_H$	21.4.3
OCSL6	Compare control register lower 6	0000 025B$_H$	21.4.3

P

PCCR0	Priority conversion control register 0	0000 012C$_H$	26.4.7
PCFD0	Priority conversion FIFO data register 0	0000 012E$_H$	26.4.9
PCIS0	Priority conversion input select register 0	0000 0130$_H$	26.4.10
PCR0	Pull-up resistor control register 0	0000 0420$_H$	15.4.5
PCR1	Pull-up resistor control register 1	0000 0421$_H$	15.4.5
PCR5	Pull-up resistor control register 5	0000 0425$_H$	15.4.5
PCR6	Pull-up resistor control register 6	0000 0426$_H$	15.4.5
PCR7	Pull-up resistor control register 7	0000 0427$_H$	15.4.5
PCR8	Pull-up resistor control register 8	0000 0428$_H$	15.4.5
PCR9	Pull-up resistor control register 9	0000 0429$_H$	15.4.5
PCRA	Pull-up resistor control register A	0000 042A$_H$	15.4.5
PCRK	Pull-up resistor control register K	0000 0434$_H$	15.4.5
PDR0	Port data register 0	0000 0000$_H$	15.4.4
PDR1	Port data register 1	0000 0001$_H$	15.4.4
PDR2	Port data register 2	0000 0002$_H$	15.4.4
PDR3	Port data register 3	0000 0003$_H$	15.4.4
PDR4	Port data register 4	0000 0004$_H$	15.4.4
PDR5	Port data register 5	0000 0005$_H$	15.4.4
PDR6	Port data register 6	0000 0006$_H$	15.4.4
PDR7	Port data register 7	0000 0007$_H$	15.4.4
PDR8	Port data register 8	0000 0008$_H$	15.4.4
PDR9	Port data register 9	0000 0009$_H$	15.4.4
PDRA	Port data register A	0000 000A$_H$	15.4.4
PDRG	Port data register G	0000 0010$_H$	15.4.4
PDRH	Port data register H	0000 0011$_H$	15.4.4

PDRK	Port data register K	0000 0014_H	15.4.4
PFNS0	Priority conversion FIFO number setting register 0	0000 $012D_H$	26.4.8
PFR0	Port function register 0	0000 $04A0_H$	15.4.2
PFR1	Port function register 1	0000 $04A1_H$	15.4.2
PFR2	Port function register 2	0000 $04A2_H$	15.4.2
PFR3	Port function register 3	0000 $04A3_H$	15.4.2
PFR4	Port function register 4	0000 $04A4_H$	15.4.2
PFR5	Port function register 5	0000 $04A5_H$	15.4.2
PFR6	Port function register 6	0000 $04A6_H$	15.4.2
PFR7	Port function register 7	0000 $04A7_H$	15.4.2
PFR8	Port function register 8	0000 $04A8_H$	15.4.2
PFRA	Port function register A	0000 $04AA_H$	15.4.2
PFRG	Port function register G	0000 $04B0_H$	15.4.2
PFRH	Port function register H	0000 $04B1_H$	15.4.2
PLLCR	PLL configuration register	0000 0514_H	4.4.4

R

RCRH0	Reload compare register upper 0	0000 $01C0_H$	25.4.1
RCRH1	Reload compare register upper 1	0000 $01D0_H$	25.4.1
RCRH2	Reload compare register upper 2	0000 $01E0_H$	25.4.1
RCRH3	Reload compare register upper 3	0000 $01F0_H$	25.4.1
RCRL0	Reload compare register lower 0	0000 $01C1_H$	25.4.1
RCRL1	Reload compare register lower 1	0000 $01D1_H$	25.4.1
RCRL2	Reload compare register lower 2	0000 $01E1_H$	25.4.1
RCRL3	Reload compare register lower 3	0000 $01F1_H$	25.4.1
RDR0	Received data register 0	0000 0064_H	28.4.5, 28.12.5
RDR1	Received data register 1	0000 $006C_H$	28.4.5, 28.12.5, 28.19.5
RDR2	Received data register 2	0000 0078_H	28.4.5, 28.12.5, 28.19.5
RDR3	Received data register 3	0000 0084_H	28.4.5, 28.12.5, 28.19.5

RDR4	Received data register 4	0000 0090_H	28.4.5, 28.12.5, 28.19.5
RDR5	Received data register 5	0000 $009C_H$	28.4.5, 28.12.5, 28.19.5
RDR6	Received data register 6	0000 $00A8_H$	28.4.5, 28.12.5, 28.19.5
RDR7	Received data register 7	0000 $00B4_H$	28.4.5, 28.12.5, 28.19.5
RDR8	Received data register 8	0000 $00D4_H$	28.4.5, 28.12.5, 28.19.5
RDR9	Received data register 9	0000 $00E4_H$	28.4.5, 28.12.5, 28.19.5
RDR10	Received data register 10	0000 $00F4_H$	28.4.5, 28.12.5, 28.19.5
RDR11	Received data register 11	0000 0104_H	28.4.5, 28.12.5, 28.19.5
RDRM0	Received data mirror register 0	0000 $00C0_H$	28.12.11
RDRM1	Received data mirror register 1	0000 $00C1_H$	28.12.11
RDRM2	Received data mirror register 2	0000 $00C2_H$	28.12.11
RDRM3	Received data mirror register 3	0000 $00C3_H$	28.12.11
RDRM4	Received data mirror register 4	0000 $00C4_H$	28.12.11
RDRM5	Received data mirror register 5	0000 $00C5_H$	28.12.11
RDRM6	Received data mirror register 6	0000 $00C6_H$	28.12.11
RDRM7	Received data mirror register 7	0000 $00C7_H$	28.12.11
RSTCR	Reset control register	0000 0481_H	9.4.2
RSTRR	Reset result register	0000 0480_H	9.4.1

S

SCCR0	Scan conversion control register 0	0000 0124_H	26.4.3
SCFD0	Scan conversion FIFO data register 0	0000 0126_H	26.4.5
SCIS00	Scan conversion input select register 00	0000 $012B_H$	26.4.6
SCIS10	Scan conversion input select register 10	0000 $012A_H$	26.4.6
SCIS20	Scan conversion input select register 20	0000 0129_H	26.4.6
SCR0	Serial control register 0	0000 0060_H	28.4.1, 28.12.1
SCR1	Serial control register 1	0000 0068_H	28.4.1, 28.12.1
SCR2	Serial control register 2	0000 0074_H	28.4.1, 28.12.1
SCR3	Serial control register 3	0000 0080_H	28.4.1, 28.12.1
SCR4	Serial control register 4	0000 $008C_H$	28.4.1, 28.12.1

SCR5	Serial control register 5	0000 0098_H	28.4.1, 28.12.1
SCR6	Serial control register 6	0000 $00A4_H$	28.4.1, 28.12.1
SCR7	Serial control register 7	0000 $00B0_H$	28.4.1, 28.12.1
SCR8	Serial control register 8	0000 $00D0_H$	28.4.1, 28.12.1
SCR9	Serial control register 9	0000 $00E0_H$	28.4.1, 28.12.1
SCR10	Serial control register 10	0000 $00F0_H$	28.4.1, 28.12.1
SCR11	Serial control register 11	0000 0100_H	28.4.1, 28.12.1
SFNS0	Scan conversion FIFO number setting register 0	0000 0125_H	26.4.4
SLIFCR	Control register	0000 2190_H	14.4.1
SLIFDRXR	DMA reception FIFO register	0000 $21A0_H$	14.4.9
SLIFRXDR	Reception FIFO data register	0000 2180_H	14.4.5
SLIFRXR	Reception register	0000 2188_H	14.4.7
SLIFRXSR	Reception FIFO status register	0000 2198_H	14.4.3
SLIFSR	Status register	0000 2194_H	14.4.2
SLIFTXDR	Transmission FIFO data register	0000 2184_H	14.4.6
SLIFTXR	Transmission register	0000 $218C_H$	14.4.8
SLIFTXSR	Transmission FIFO status register	0000 $219C_H$	14.4.4
SLPRR	Sleep rate configuration register	0000 0483_H	8.3.2
SMR0	Serial mode register 0	0000 0061_H	28.4.2, 28.12.2
SMR1	Serial mode register 1	0000 0069_H	28.4.2, 28.12.2, 28.19.2
SMR2	Serial mode register 2	0000 0075_H	28.4.2, 28.12.2, 28.19.2
SMR3	Serial mode register 3	0000 0081_H	28.4.2, 28.12.2, 28.19.2
SMR4	Serial mode register 4	0000 $008D_H$	28.4.2, 28.12.2, 28.19.2
SMR5	Serial mode register 5	0000 0099_H	28.4.2, 28.12.2, 28.19.2
SMR6	Serial mode register 6	0000 $00A5_H$	28.4.2, 28.12.2, 28.19.2
SMR7	Serial mode register 7	0000 $00B1_H$	28.4.2, 28.12.2, 28.19.2
SMR8	Serial mode register 8	0000 $00D1_H$	28.4.2, 28.12.2, 28.19.2
SMR9	Serial mode register 9	0000 $00E1_H$	28.4.2, 28.12.2, 28.19.2
SMR10	Serial mode register 10	0000 $00F1_H$	28.4.2, 28.12.2, 28.19.2
SMR11	Serial mode register 11	0000 0101_H	28.4.2, 28.12.2, 28.19.2
SSEL0123	Serial mode select register 0123	0000 $00C8_H$	28.12.10

SSEL4567	Serial mode select register 4567	0000 00CA$_H$	28.12.10
SSR0	Serial status register 0	0000 0062$_H$	28.4.3, 28.12.3
SSR1	Serial status register 1	0000 006A$_H$	28.4.3, 28.12.3, 28.19.4
SSR2	Serial status register 2	0000 0076$_H$	28.4.3, 28.12.3, 28.19.4
SSR3	Serial status register 3	0000 0082$_H$	28.4.3, 28.12.3, 28.19.4
SSR4	Serial status register 4	0000 008E$_H$	28.4.3, 28.12.3, 28.19.4
SSR5	Serial status register 5	0000 009A$_H$	28.4.3, 28.12.3, 28.19.4
SSR6	Serial status register 6	0000 00A6$_H$	28.4.3, 28.12.3, 28.19.4
SSR7	Serial status register 7	0000 00B2$_H$	28.4.3, 28.12.3, 28.19.4
SSR8	Serial status register 8	0000 00D2$_H$	28.4.3, 28.12.3, 28.19.4
SSR9	Serial status register 9	0000 00E2$_H$	28.4.3, 28.12.3, 28.19.4
SSR10	Serial status register 10	0000 00F2$_H$	28.4.3, 28.12.3, 28.19.4
SSR11	Serial status register 11	0000 0102$_H$	28.4.3, 28.12.3, 28.19.4
STBCR	Standby mode control register	0000 0482$_H$	8.3.1
STMCR	Sub timer control register	0000 0513$_H$	7.3.1

T

TCCSH0	Timer status control register upper 0	0000 0208$_H$	19.4.4
TCCSH1	Timer status control register upper 1	0000 0268$_H$	19.4.4
TCCSL0	Timer status control register lower 0	0000 0209$_H$	19.4.4
TCCSL1	Timer status control register lower 1	0000 0269$_H$	19.4.4
TCDT0	Timer data register 0	0000 0204$_H$	19.4.3
TCDT1	Timer data register 1	0000 0264$_H$	19.4.3
TDR0	Transmitted data register 0	0000 0064$_H$	28.4.5, 28.12.5
TDR1	Transmitted data register 1	0000 006C$_H$	28.4.5, 28.12.5, 28.19.5
TDR2	Transmitted data register 2	0000 0078$_H$	28.4.5, 28.12.5, 28.19.5
TDR3	Transmitted data register 3	0000 0084$_H$	28.4.5, 28.12.5, 28.19.5
TDR4	Transmitted data register 4	0000 0090$_H$	28.4.5, 28.12.5, 28.19.5
TDR5	Transmitted data register 5	0000 009C$_H$	28.4.5, 28.12.5, 28.19.5
TDR6	Transmitted data register 6	0000 00A8$_H$	28.4.5, 28.12.5, 28.19.5

TDR7	Transmitted data register 7	0000 00B4$_H$	28.4.5, 28.12.5, 28.19.5
TDR8	Transmitted data register 8	0000 00D4$_H$	28.4.5, 28.12.5, 28.19.5
TDR9	Transmitted data register 9	0000 00E4$_H$	28.4.5, 28.12.5, 28.19.5
TDR10	Transmitted data register 10	0000 00F4$_H$	28.4.5, 28.12.5, 28.19.5
TDR11	Transmitted data register 11	0000 0104$_H$	28.4.5, 28.12.5, 28.19.5
TDRM0	Transmitted data mirror register 0	0000 00C0$_H$	28.12.11
TDRM1	Transmitted data mirror register 1	0000 00C1$_H$	28.12.11
TDRM2	Transmitted data mirror register 2	0000 00C2$_H$	28.12.11
TDRM3	Transmitted data mirror register 3	0000 00C3$_H$	28.12.11
TDRM4	Transmitted data mirror register 4	0000 00C4$_H$	28.12.11
TDRM5	Transmitted data mirror register 5	0000 00C5$_H$	28.12.11
TDRM6	Transmitted data mirror register 6	0000 00C6$_H$	28.12.11
TDRM7	Transmitted data mirror register 7	0000 00C7$_H$	28.12.11
TMCSR0	Timer control status register 0	0000 004E$_H$	22.4.1
TMCSR1	Timer control status register 1	0000 0056$_H$	22.4.1
TMCSR2	Timer control status register 2	0000 005E$_H$	22.4.1
TMR0	16-bit timer register 0	0000 004A$_H$	22.4.3
TMR1	16-bit timer register 1	0000 0052$_H$	22.4.3
TMR2	16-bit timer register 2	0000 005A$_H$	22.4.3
TMRLRA0	16-bit timer reload register A0	0000 0048$_H$	22.4.2
TMRLRA1	16-bit timer reload register A1	0000 0050$_H$	22.4.2
TMRLRA2	16-bit timer reload register A2	0000 0058$_H$	22.4.2
TMSP	Time stamp register	0000 213C$_H$	31.3.6

U

UCCR	USB clock setting register	0000 051C$_H$	29.3.1
UDCC	UDC control register	0000 2121$_H$	31.3.3
UDCIE	UDC interrupt request register	0000 2140$_H$	31.3.8
UDCRH0	Up-down count register upper 0	0000 01C2$_H$	25.4.2
UDCRH1	Up-down count register upper 1	0000 01D2$_H$	25.4.2

UDCRH2	Up-down count register upper 2	0000 01E2$_H$	25.4.2
UDCRH3	Up-down count register upper 3	0000 01F2$_H$	25.4.2
UDCRL0	Up-down count register lower 0	0000 01C3$_H$	25.4.2
UDCRL1	Up-down count register lower 1	0000 01D3$_H$	25.4.2
UDCRL2	Up-down count register lower 2	0000 01E3$_H$	25.4.2
UDCRL3	Up-down count register lower 3	0000 01F3$_H$	25.4.2
UDCS	UDC status register	0000 2141$_H$	31.3.7
USBEN	USB request register	0000 21A6$_H$	31.3.2
USBSEL	USB select register	0000 21A5$_H$	31.3.1

W

WCCR	Watch counter control register	0000 051A$_H$	18.3.2
WCRD	Watch counter read register	0000 0518$_H$	18.3.3
WCRL	Watch counter reload register	0000 0519$_H$	18.3.1
WDTCPR0	Watchdog timer clear pattern register 0	0000 003D$_H$	17.3.2
WDTCR0	Watchdog timer control register 0	0000 003C$_H$	17.3.1
WRAR00	Wild register address register 00	0000 0380$_H$	37.3.1
WRAR01	Wild register address register 01	0000 0388$_H$	37.3.1
WRAR02	Wild register address register 02	0000 0390$_H$	37.3.1
WRAR03	Wild register address register 03	0000 0398$_H$	37.3.1
WRAR04	Wild register address register 04	0000 03A0$_H$	37.3.1
WRAR05	Wild register address register 05	0000 03A8$_H$	37.3.1
WRAR06	Wild register address register 06	0000 03B0$_H$	37.3.1
WRAR07	Wild register address register 07	0000 03B8$_H$	37.3.1
WRAR08	Wild register address register 08	0000 03C0$_H$	37.3.1
WRAR09	Wild register address register 09	0000 03C8$_H$	37.3.1
WRAR10	Wild register address register 10	0000 03D0$_H$	37.3.1
WRAR11	Wild register address register 11	0000 03D8$_H$	37.3.1
WRAR12	Wild register address register 12	0000 03E0$_H$	37.3.1

WRAR13	Wild register address register 13	0000 $03E8_H$	37.3.1
WRAR14	Wild register address register 14	0000 $03F0_H$	37.3.1
WRAR15	Wild register address register 15	0000 $03F8_H$	37.3.1
WRDR00	Wild register data register 00	0000 0384_H	37.3.2
WRDR01	Wild register data register 01	0000 $038C_H$	37.3.2
WRDR02	Wild register data register 02	0000 0394_H	37.3.2
WRDR03	Wild register data register 03	0000 $039C_H$	37.3.2
WRDR04	Wild register data register 04	0000 $03A4_H$	37.3.2
WRDR05	Wild register data register 05	0000 $03AC_H$	37.3.2
WRDR06	Wild register data register 06	0000 $03B4_H$	37.3.2
WRDR07	Wild register data register 07	0000 $03BC_H$	37.3.2
WRDR08	Wild register data register 08	0000 $03C4_H$	37.3.2
WRDR09	Wild register data register 09	0000 $03CC_H$	37.3.2
WRDR10	Wild register data register 10	0000 $03D4_H$	37.3.2
WRDR11	Wild register data register 11	0000 $03DC_H$	37.3.2
WRDR12	Wild register data register 12	0000 $03E4_H$	37.3.2
WRDR13	Wild register data register 13	0000 $03EC_H$	37.3.2
WRDR14	Wild register data register 14	0000 $03F4_H$	37.3.2
WRDR15	Wild register data register 15	0000 $03FC_H$	37.3.2
WREN	Wild register enable register	0000 $033A_H$	37.3.3

APPENDIX C Interrupt Vectors

This appendix explains the interrupt vector table for this series. This table specifies how interrupt sources are assigned to interrupt vectors and interrupt control registers (ICR00 to ICR47).

Interrupt Source (Peripheral Function)	Number		Interrupt level setting registers	Offset	TBR Initial address
	Dec.	Hex.			
Reset	0	00	-	$3FC_H$	000F $FFFC_H$
Reserved for system	1	01	-	$3F8_H$	000F $FFF8_H$
Reserved for system	2	02	-	$3F4_H$	000F $FFF4_H$
Reserved for system	3	03	-	$3F0_H$	000F $FFF0_H$
Reserved for system	4	04	-	$3EC_H$	000F $FFEC_H$
Reserved for system	5	05	-	$3E8_H$	000F $FFE8_H$
Reserved for system	6	06	-	$3E4_H$	000F $FFE4_H$
Reserved for system	7	07	-	$3E0_H$	000F $FFE0_H$
Reserved for system	8	08	-	$3DC_H$	000F $FFDC_H$
INTE instruction	9	09	-	$3D8_H$	000F $FFD8_H$
Reserved for system	10	0A	-	$3D4_H$	000F $FFD4_H$
Reserved for system	11	0B	-	$3D0_H$	000F $FFD0_H$
Step trace trap	12	0C	-	$3CC_H$	000F $FFCC_H$
Reserved for system	13	0D	-	$3C8_H$	000F $FFC8_H$
Undefined instruction exception	14	0E	-	$3C4_H$	000F $FFC4_H$
-	15	0F	Always 15(FH)	$3C0_H$	000F $FFC0_H$
External interrupt request ch.0 to ch.7	16	10	ICR00	$3BC_H$	000F $FFBC_H$
External interrupt request ch.8 to ch.15	17	11	ICR01	$3B8_H$	000F $FFB8_H$
External interrupt request ch.16 to ch.23	18	12	ICR02	$3B4_H$	000F $FFB4_H$
External interrupt request ch.24 to ch.31	19	13	ICR03	$3B0_H$	000F $FFB0_H$
16-bit reload timer ch.0 to ch.2	20	14	ICR04	$3AC_H$	000F $FFAC_H$
Reception interrupt request from UART/CSIO ch.0	21	15	ICR05	$3A8_H$	000F $FFA8_H$

Interrupt Source (Peripheral Function)	Number		Interrupt level setting registers	Offset	TBR Initial address
	Dec.	Hex.			
Transmission interrupt request from UART/CSIO ch.0 Transmission bus idle interrupt request from UART/CSIO ch.0	22	16	ICR06	$3A4_H$	000F FFA4$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.1	23	17	ICR07	$3A0_H$	000F FFA0$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.1 Transmission bus idle interrupt request from UART/CSIO ch.1	24	18	ICR08	$39C_H$	000F FF9C$_H$
Status interrupt request from I^{2}C ch.1	25	19	ICR09	398_H	000F FF98$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.2	26	1A	ICR10	394_H	000F FF94$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.2 Transmission bus idle interrupt request from UART/CSIO ch.2	27	1B	ICR11	390_H	000F FF90$_H$
Status interrupt request from I^{2}C ch.2	28	1C	ICR12	$38C_H$	000F FF8C$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.3	29	1D	ICR13	388_H	000F FF88$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.3 Transmission bus idle interrupt request from UART/CSIO ch.3 Status interrupt request from I^{2}C ch.3	30	1E	ICR14	384_H	000F FF84$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.4	31	1F	ICR15	380_H	000F FF80$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.4 Transmission bus idle interrupt request from UART/CSIO ch.4 Status interrupt request from I^{2}C ch.4	32	20	ICR16	$37C_H$	000F FF7C$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.5	33	21	ICR17	378_H	000F FF78$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.5 Transmission bus idle interrupt request from UART/CSIO ch.5 Status interrupt request from I^{2}C ch.5	34	22	ICR18	374_H	000F FF74$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.6	35	23	ICR19	370_H	000F FF70$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.6 Transmission bus idle interrupt request from UART/CSIO ch.6 Status interrupt request from I^{2}C ch.6	36	24	ICR20	$36C_H$	000F FF6C$_H$
Reception interrupt request from UART/CSIO/I^{2}C ch.7 32-bit input capture ch.4 to ch.7	37	25	ICR21	368_H	000F FF68$_H$
Transmission interrupt request from UART/CSIO/I^{2}C ch.7 Transmission bus idle interrupt request from UART/CSIO ch.7 Status interrupt request from I^{2}C ch.7 32-bit output compare ch.4 to ch.7	38	26	ICR22	364_H	000F FF64$_H$

Interrupt Source (Peripheral Function)	Number		Interrupt level setting registers	Offset	TBR Initial address
	Dec.	Hex.			
Reception interrupt request from UART/CSIO/I^{2}C ch.8 to ch.11 Transmission interrupt request from UART/CSIO/I^{2}C ch.8 to ch.11 Transmission bus idle interrupt request from UART/CSIO ch.8 to ch.11 Transmission FIFO interrupt request from UART/CSIO/I^{2}C ch.8 to ch.11 Status interrupt request from I^{2}C ch.8 to ch.11	39	27	ICR23	360_H	000F FF60$_H$
16-bit up/down counter ch.0 to ch.3	40	28	ICR24	$35C_H$	000F FF5C$_H$
Main timer/sub timer/watch counter	41	29	ICR25	358_H	000F FF58$_H$
10-bit A/D converter - Scanning conversion interrupt request - Priority conversion interrupt request - FIFO overrun interrupt request - Conversion result comparison interrupt request	42	2A	ICR26	354_H	000F FF54$_H$
32-bit free-run timer ch.0, ch.1	43	2B	ICR27	350_H	000F FF50$_H$
32-bit input capture ch.0 to ch.3	44	2C	ICR28	$34C_H$	000F FF4C$_H$
32-bit output compare ch.0 to ch.3	45	2D	ICR29	348_H	000F FF48$_H$
Base timer ch.0	46	2E	ICR30	344_H	000F FF44$_H$
Base timer ch.1	47	2F	ICR31	340_H	000F FF40$_H$
Base timer ch.2	48	30	ICR32	$33C_H$	000F FF3C$_H$
Base timer ch.3	49	31	ICR33	338_H	000F FF38$_H$
Base timer ch.4, ch.5	50	32	ICR34	334_H	000F FF34$_H$
Base timer ch.6, ch.7	51	33	ICR35	330_H	000F FF30$_H$
Base timer ch.8, ch.9	52	34	ICR36	$32C_H$	000F FF2C$_H$
Base timer ch.10, ch.11	53	35	ICR37	328_H	000F FF28$_H$
Base timer ch.12/USB function (End point 1 to 5 for DRQ)	54	36	ICR38	329_H	000F FF24$_H$
Base timer ch.13/USB function (End point 0 for DRQI, DRQO and each status)/USB Mini-HOST (each status)	55	37	ICR39	320_H	000F FF20$_H$
Base timer ch.14, ch.15/slave interface	56	38	ICR40	$31C_H$	000F FF1C$_H$
DMA Controller (DMAC) ch.0	57	39	ICR41	318_H	000F FF18$_H$
DMA Controller (DMAC) ch.1	58	3A	ICR42	314_H	000F FF14$_H$
DMA Controller (DMAC) ch.2	59	3B	ICR43	310_H	000F FF10$_H$
DMA Controller (DMAC) ch.3	60	3C	ICR44	$30C_H$	000F FF0C$_H$
DMA Controller (DMAC) ch.4 to ch.7	61	3D	ICR45	308_H	000F FF08$_H$

Interrupt Source (Peripheral Function)	Number		Interrupt level setting registers	Offset	TBR Initial address
	Dec.	Hex.			
System reserved	62	3E	ICR46	304_H	000F FF04$_H$
Delay interrupt	63	3F	ICR47	300_H	000F FF00$_H$
Reserved for system (used by REALOS)	64	40	-	$2FC_H$	000F FEFC$_H$
Reserved for system (used by REALOS)	65	41	-	$2F8_H$	000F FEF8$_H$
Used by the INT instruction	66 to 255	42 to FF	-	$2F4_H$ to 000_H	000F FEF4$_H$ to 000F FC00$_H$

* : USB interrupt source

Number		USB interrupt source	Details
Decimal	Hexadecimal		
54	36	USB function (DRQ of End Point 1 to 5)	DRQ (End Point1 to 5)
55	37	USB function (DRQI, DRQO of End Point 0 and each status)	DRQI, DRQO, SPK, SUSP, SOF, BRST, CONF, WKUP
		USB Mini-HOST (Each status)	DIRQ, URIRQ, RWKIRQ, CNNIRQ, SOFIRQ, CMPIRQ

APPENDIX D Pin State in Each CPU State

This appendix lists the pin state for each CPU state.

■ Pin state

The terms used in the table have the meanings as follows.

- $\overline{\text{INIT}}$ = "L" Period

  This is the period in which the $\overline{\text{INIT}}$ pin is at the "L" level.
- $\overline{\text{INIT}}$ = "H" Period

  The $\overline{\text{INIT}}$ pin is in the state immediately following a transition from the "L" to "H" level.
- Input enabled

  This indicates that the input function can be used.
- Input disabled

  This indicates that the input function cannot be used.
- Output Hi-Z

  The pin is placed in Hi-Z by preventing the transistor from driving the pin.
- Last state maintained

  The output state immediately before this mode is entered is maintained.

  If any built-in peripheral function is active, output is performed according to that peripheral function.

  Any output that is performed for operation such as for a port is maintained.
- Internal input "0" fixed

  External input is cut off at the input gate immediately next to the pin, and "0" is sent to the CPU.
- Input enabled when the selection of interrupt function is enabled

  The pin functions are set for an external interrupt request input pin to enable input only when external interrupt requests are enabled.

● List of pin status

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
$\overline{INIT}$	$\overline{INIT}$	—	—	Input enabled	Input enabled	Input enabled
X0	X0	Input enabled	Input enabled		Hi-Z or Input enabled	Hi-Z or Input enabled
X1	X1	Input enabled	Input enabled		“H” output or Input enabled	“H”output or Input enabled
X0A	X0A (When $\overline{INIT}$ input, see PK1. When port selected, input disabled)	Input disabled	Input disabled		Hi-Z or Input enabled	Hi-Z or Input enabled
X1A	X1A (When $\overline{INIT}$ input, see PK0. When port selected, input disabled)	Input disabled	Input disabled		“H”output or Input enabled	“H”output or Input enabled
MD0	MD0	Input enabled	Input enabled		Input enabled	Input enabled
MD1	MD1	Input enabled	Input enabled			
P00	P00/D00/TIOA0/SOUT0_1/IN0	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0”
P01	P01/D01/TIOB0/SIN0_1/IN1					
P02	P02/D02/TIOA1/SCK0_1/IN2					
P03	P03/D03/TIOB1/IN3					
P04	P04/D04/TIOA2/SOUT1/IN4					
P05	P05/D05/TIOB2/SIN1/IN5					
P06	P06/D06/TIOA3/SCK1/IN6					
P07	P07/D07/TIOB3/IN7					

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
P10	P10/D08/TIOA4/SOUT2/AIN0/ INT0	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
P11	P11/D09/TIOB4/SIN2/BIN0/INT1					
P12	P12/D10/TIOA5/SCK2/ZIN0/ INT2					
P13	P13/D11/TIOB5/INT3					
P14	P14/D12/TIOA6/SOUT3/AIN1/ INT4					
P15	P15/D13/TIOB6/SIN3/BIN1/INT5					
P16	P16/D14/TIOA7/SCK3/ZIN1/ INT6					
P17	P17/D15/TIOB7/INT7					
P20	P20/A00/TIOA8/SOUT4/AIN2	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0”
P21	P21/A01/TIOB8/SIN4/BIN2					
P22	P22/A02/TIOA9/SCK4/ZIN2					
P23	P23/A03/TIOB9					
P24	P24/A04/TIOA10/SOUT5/AIN3/ OUT0					
P25	P25/A05/TIOB10/SIN5/BIN3/ OUT1					
P26	P26/A06/TIOA11/SCK5/ZIN3/ OUT2					
P27	P27/A07/TIOB11/OUT3					

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
P30	P30/A08/TIOA12/SOUT6/INT8/SIFD6_1	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0" Input enabled when interrupt function selected and enabled
P31	P31/A09/TIOB12/SIN6/INT9/SIFD7_1					
P32	P32/A10/TIOA13/SCK6/INT10/SIFD8_1					
P33	P33/A11/TIOB13/INT11/SIFD9_1					
P34	P34/A12/TIOA14/SOUT7/OUT4/INT12/SIFD10_1					
P35	P35/A13/TIOB14/SIN7/OUT5/INT13/SIFD11_1					
P36	P36/A14/TIOA15/SCK7/OUT6/INT14/SIFD12_1					
P37	P37/A15/TIOB15/OUT7/INT15/SIFD13_1					
P40	P40/A16/SOUT8/SIFD14_1	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
P41	P41/A17/SIN8/SIFD15_1					
P42	P42/A18/SCK8/$\overline{SIFCS\_1}$					
P43	P43/A19/SIFA_1					
P44	P44/A20/SOUT9/$\overline{SIFWR\_1}$					
P45	P45/A21/SIN9/$\overline{SIFRD\_1}$					
P46	P46/A22/SCK9/$\overline{SIFDRQR\_1}$					
P47	P47/A23/$\overline{SIFDRQT\_1}$					

<table>
<tr><th rowspan="2">Pin Name</th><th rowspan="2">Function Name</th><th colspan="2">During initialization</th><th rowspan="2">In sleep mode</th><th colspan="2">In stop mode</th></tr>
<tr><th>$\overline{\text{INIT}}$ = "L" Period</th><th>$\overline{\text{INIT}}$ = "H" Period</th><th>HIZ = 0</th><th>HIZ = 1</th></tr>
<tr><td>P50</td><td>P50/$\overline{\text{CS0}}$/SOUT10/AIN0_1</td><td rowspan="9">Output Hi-Z</td><td rowspan="9">Output Hi-Z/ Input enabled</td><td rowspan="9">Maintain previous state</td><td rowspan="9">Maintain previous state</td><td rowspan="3">Output Hi-Z/ Internal input fixed at “0”</td></tr>
<tr><td>P51</td><td>P51/$\overline{\text{CS1}}$/SIN10/BIN0_1</td></tr>
<tr><td>P52</td><td>P52/$\overline{\text{CS2}}$/SCK10/ZIN0_1</td></tr>
<tr><td>P53</td><td>P53/$\overline{\text{CS3}}$/FRCK1/INT21_2</td><td>Output Hi-Z/ Internal input fixed at “0”

Input enabled when interrupt function selected and enabled</td></tr>
<tr><td>P54</td><td>P54/$\overline{\text{AS}}$/SOUT11/AIN1_1</td><td rowspan="4">Output Hi-Z/ Internal input fixed at “0”</td></tr>
<tr><td>P55</td><td>P55/$\overline{\text{RD}}$/SIN11/BIN1_1/ADTRG0</td></tr>
<tr><td>P56</td><td>P56/$\overline{\text{WR0}}$/SCK11/ZIN1_1/FRCK0</td></tr>
<tr><td>P57</td><td>P57/$\overline{\text{WR1}}$</td></tr>
</table>

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
P60	P60/RDY/AIN2_1	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0”
P61	P61/SYSCLK/BIN2_1					
P62	P62/DREQ0/ZIN2_1					
P63	P63/DACK0/FRCK1_1/INT22_2					Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
P64	P64/DEOP0/AIN3_1					Output Hi-Z/ Internal input fixed at “0”
P65	P65/DREQ1/BIN3_1/ADTRG0_1					
P66	P66/DACK1/ZIN3_1/FRCK0_1					
P67	P67/DEOP1/INT23_2					Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
P70	P70/AN0/OUT0_1/INT16	Output Hi-Z	Output Hi-Z/ Input disabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
P71	P71/AN1/OUT1_1/INT17					
P72	P72/AN2/TMO0/OUT2_1/INT18/ SIFD0					
P73	P73/AN3/TMO1/OUT3_1/INT19/ SIFD1					
P74	P74/AN4/TMO2/OUT4_1/INT20/ SIFD2					
P75	P75/AN5/SOUT0/TMI0/OUT5_1/ INT21/SIFD3					
P76	P76/AN6/SIN0/TMI1/OUT6_1/ INT22/SIFD4					
P77	P77/AN7/SCK0/TMI2/OUT7_1/ INT23/SIFD5					
P80	P80/AN8/IN0_1/INT24/SIFD6	Output Hi-Z	Output Hi-Z/ Input disabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
P81	P81/AN9/IN1_1/INT25/SIFD7					
P82	P82/AN10/IN2_1/INT26/SIFD8					
P83	P83/AN11/IN3_1/INT27/SIFD9					
P84	P84/AN12/IN4_1/INT28/SIFD10					
P85	P85/AN13/IN5_1/INT29/SIFD11					
P86	P86/AN14/IN6_1/INT30/SIFD12					
P87	P87/AN15/IN7_1/INT31/SIFD13					
P90	P90/DA0	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0”
P91	P91/DA1					
P92	P92/DA2					

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
PA0	PA0/AN16/INT16_1/SIFD14	Output Hi-Z	Output Hi-Z/ Input disabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
PA1	PA1/AN17/INT17_1/SIFD15					
PA2	PA2/AN18/TMO0_1/INT18_1/ $\overline{SIFCS}$					
PA3	PA3/AN19/TMO1_1/INT19_1/ $\overline{SIFA}$					
PA4	PA4/AN20/TMO2_1/INT20_1					
PA5	PA5/AN21/TMI0_1/INT21_1					
PA6	PA6/AN22/TMI1_1/INT22_1					
PA7	PA7/AN23/TMI2_1/INT23_1					
PG0	PG0/DREQ2/TIOA0_1/SOUT0_2/ IN0_2	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0”
PG1	PG1/DACK2/TIOB0_1/SIN0_2/ IN1_2					
PG2	PG2/DEOP2/TIOA1_1/SCK0_2/ IN2_2					
PG3	PG3/DREQ3/TIOB1_1/IN3_2					
PG4	PG4/DACK3/TIOA2_1/SOUT1_1/ IN4_2					
PG5	PG5/DEOP3/TIOB2_1/SIN1_1/ IN5_2					
PG6	PG6/TIOA3_1/SCK1_1/IN6_2					
PG7	PG7/TIOB3_1/IN7_2					
PH0	PH0/TIOA4_1/SOUT2_1/ AIN0_2/INT0_1	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at “0” Input enabled when interrupt function selected and enabled
PH1	PH1/TIOB4_1/SIN2_1/BIN0_2/ INT1_1					
PH2	PH2/TIOA5_1/SCK2_1/ZIN0_2/ INT2_1					
PH3	PH3/TIOB5_1/INT3_1					

Pin Name	Function Name	During initialization		In sleep mode	In stop mode	
		$\overline{INIT}$ = "L" Period	$\overline{INIT}$ = "H" Period		HIZ = 0	HIZ = 1
PK0	PK0	Output Hi-Z	Output Hi-Z/ Internal input fixed at "0"	Maintain previous state	Maintain previous state	Output Hi-Z/ Internal input fixed at "0"
PK1	PK1					
PK2	PK2/ADTRG0_2		Output Hi-Z/ Input enabled			
PK3	PK3/ADTRG0_3					
UDP	UDP (USB)	Output Hi-Z	Output Hi-Z/ Input enabled	Maintain previous state/Input enabled	Maintain previous state	Maintain previous state
UDM	UDM (USB)					

● List of pin status (serial write mode)

Pin name	Function	During initialization	During asynchronous write operation	During synchronous write operation
		$\overline{\text{INIT}}$ = "L"	$\overline{\text{INIT}}$ = "H"	
$\overline{\text{INIT}}$	$\overline{\text{INIT}}$	—	—	—
X0	X0	Input enabled	Input enabled	Input enabled
X1	X1	Input enabled	Input enabled	Input enabled
X0A	X0A (When $\overline{\text{INIT}}$ input, see PK1. When port selected, input disabled)	Input disabled	Input disabled	Input disabled
X1A	X1A (When $\overline{\text{INIT}}$ input, see PK0. When port selected, input disabled)	Input disabled	Input disabled	Input disabled
MD0	MD0	Input enabled	Input enabled	Input enabled
MD1	MD1	Input enabled	Input enabled	Input enabled
P00	P00/D00/TIOA0/SOUT0_1/IN0	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P01	P01/D01/TIOB0/SIN0_1/IN1			
P02	P02/D02/TIOA1/SCK0_1/IN2			
P03	P03/D03/TIOB1/IN3			
P04	P04/D04/TIOA2/SOUT1/IN4			
P05	P05/D05/TIOB2/SIN1/IN5			
P06	P06/D06/TIOA3/SCK1/IN6			
P07	P07/D07/TIOB3/IN7			

Pin name	Function	During initialization	During asynchronous write operation	During synchronous write operation
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"	
P10	P10/D08/TIOA4/SOUT2/AIN0/ INT0	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P11	P11/D09/TIOB4/SIN2/BIN0/INT1			
P12	P12/D10/TIOA5/SCK2/ZIN0/ INT2			
P13	P13/D11/TIOB5/INT3			
P14	P14/D12/TIOA6/SOUT3/AIN1/ INT4			
P15	P15/D13/TIOB6/SIN3/BIN1/INT5			
P16	P16/D14/TIOA7/SCK3/ZIN1/ INT6			
P17	P17/D15/TIOB7/INT7			
P20	P20/A00/TIOA8/SOUT4/AIN2	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P21	P21/A01/TIOB8/SIN4/BIN2			
P22	P22/A02/TIOA9/SCK4/ZIN2			
P23	P23/A03/TIOB9			
P24	P24/A04/TIOA10/SOUT5/AIN3/ OUT0			
P25	P25/A05/TIOB10/SIN5/BIN3/ OUT1			
P26	P26/A06/TIOA11/SCK5/ZIN3/ OUT2			
P27	P27/A07/TIOB11/OUT3			
P30	P30/A08/TIOA12/SOUT6/INT8	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P31	P31/A09/TIOB12/SIN6/INT9			
P32	P32/A10/TIOA13/SCK6/INT10			
P33	P33/A11/TIOB13/INT11			
P34	P34/A12/TIOA14/SOUT7/OUT4/ INT12			
P35	P35/A13/TIOB14/SIN7/OUT5/ INT13			
P36	P36/A14/TIOA15/SCK7/OUT6/ INT14			
P37	P37/A15/TIOB15/OUT7/INT15			

Pin name	Function	During initialization	During asynchronous write operation	During synchronous write operation
		$\overline{INIT}$ = "L"	$\overline{INIT}$ = "H"	
P40	P40/A16/SOUT8	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P41	P41/A17/SIN8			
P42	P42/A18/SCK8			
P43	P43/A19			
P44	P44/A20/SOUT9			
P45	P45/A21/SIN9			
P46	P46/A22/SCK9			
P47	P47/A23			
P50	P50/$\overline{CS0}$/SOUT10/AIN0_1	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P51	P51/$\overline{CS1}$/SIN10/BIN0_1			
P52	P52/$\overline{CS2}$/SCK10/ZIN0_1			
P53	P53/$\overline{CS3}$/FRCK1/INT21_2			
P54	P54/$\overline{AS}$/SOUT11/AIN1_1			
P55	P55/$\overline{RD}$/SIN11/BIN1_1/ADTRG0			
P56	P56/$\overline{WR0}$/SCK11/ZIN1_1/ FRCK0			
P57	P57/$\overline{WR1}$			
P60	P60/RDY/AIN2_1	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P61	P61/SYSCLK/BIN2_1			
P62	P62/DREQ0/ZIN2_1			
P63	P63/DACK0/FRCK1_1/INT22_2			
P64	P64/DEOP0/AIN3_1			
P65	P65/DREQ1/BIN3_1/ADTRG0_1			
P66	P66/DACK1/ZIN3_1/FRCK0_1			
P67	P67/DEOP1/INT23_2			
P70	P70/AN0/OUT0_1/INT16	Output Hi-Z	Output Hi-Z/ Input disabled	Output Hi-Z/ Input disabled
P71	P71/AN1/OUT1_1/INT17			
P72	P72/AN2/TMO0/OUT2_1/INT18			
P73	P73/AN3/TMO1/OUT3_1/INT19			
P74	P74/AN4/TMO2/OUT4_1/INT20			

Pin name	Function	During initialization	During asynchronous write operation	During synchronous write operation
		$\overline{\text{INIT}}$ = "L"	$\overline{\text{INIT}}$ = "H"	
P75	P75/AN5/SOUT0/TMI0/OUT5_1/ INT21	Output Hi-Z/ Input enabled	Output	Output
P76	P76/AN6/SIN0/TMI1/OUT6_1/ INT22	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P77	P77/AN7/SCK0/TMI2/OUT7_1/ INT23		Output Hi-Z/ Input disabled	Output Hi-Z/ Input enabled
P80	P80/AN8/IN0_1/INT24	Output Hi-Z	Output Hi-Z/ Input disabled	Output Hi-Z/ Input disabled
P81	P81/AN9/IN1_1/INT25			
P82	P82/AN10/IN2_1/INT26			
P83	P83/AN11/IN3_1/INT27			
P84	P84/AN12/IN4_1/INT28			
P85	P85/AN13/IN5_1/INT29			
P86	P86/AN14/IN6_1/INT30			
P87	P87/AN15/IN7_1/INT31			
P90	P90/DA0	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
P91	P91/DA1			
P92	P92/DA2			
PA0	PA0/AN16/INT16_1	Output Hi-Z	Output Hi-Z/ Input disabled	Output Hi-Z/ Input disabled
PA1	PA1/AN17/INT17_1			
PA2	PA2/AN18/TMO0_1/INT18_1			
PA3	PA3/AN19/TMO1_1/INT19_1			
PA4	PA4/AN20/TMO2_1/INT20_1			
PA5	PA5/AN21/TMI0_1/INT21_1			
PA6	PA6/AN22/TMI1_1/INT22_1			
PA7	PA7/AN23/TMI2_1/INT23_1			

Pin name	Function	During initialization	During asynchronous write operation	During synchronous write operation
		$\overline{\text{INIT}}$ = "L"	$\overline{\text{INIT}}$ = "H"	
PG0	PG0/DREQ2/TIOA0_1/ SOUT0_2/IN0_2	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
PG1	PG1/DACK2/TIOB0_1/SIN0_2/ IN1_2			
PG2	PG2/DEOP2/TIOA1_1/SCK0_2/ IN2_2			
PG3	PG3/DREQ3/TIOB1_1/IN3_2			
PG4	PG4/DACK3/TIOA2_1/ SOUT1_1/IN4_2			
PG5	PG5/DEOP3/TIOB2_1/SIN1_1/ IN5_2			
PG6	PG6/TIOA3_1/SCK1_1/IN6_2			
PG7	PG7/TIOB3_1/IN7_2			
PH0	PH0/TIOA4_1/SOUT2_1/ AIN0_2/INT0_1	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
PH1	PH1/TIOB4_1/SIN2_1/BIN0_2/ INT1_1			
PH2	PH2/TIOA5_1/SCK2_1/ZIN0_2/ INT2_1			
PH3	PH3/TIOB5_1/INT3_1			
PK0	PK0	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
PK1	PK1			
PK2	PK2/ADTRG0_2			
PK3	PK3/ADTRG0_3			
UDP	UDP (USB)	Output Hi-Z	Output Hi-Z/ Input enabled	Output Hi-Z/ Input enabled
UDM	UDM (USB)			

APPENDIX E Lists of Instructions

This section lists and maps instructions for the FR80 family CPUs.

E.1 Instruction List

This section explains the symbols used in the instruction tables and instruction rules.

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Rj	A	A6	1	CCCC		Ri + Rj -> Rj	
*ADD #s5, Rj	C	A4	1	CCCC		Ri + s5 -> Ri	
-	-	-	-	-	O	-	
-	-	-	-	-		-	
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

(1) The instruction name is shown.

Instructions marked with* are extended instructions implemented either by extending existing instructions or by coding from scratch using the assembler, which are not native to the CPU.

(2) A specifiable Addressing mode is shown in the operand by the sign.

Please refer to-the sign of the Addressing mode (next item) for the meaning of the sign.

(3) The instruction format is shown.

(4) The hexadecimal number is displayed to the instruction code (Not written in assembler extended instructions).

(5) The number of machine cycles is shown.

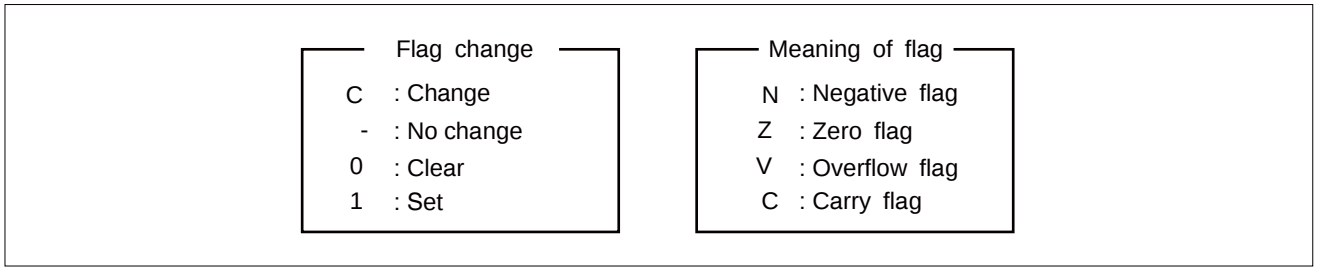
a: It is a memory access cycle, and there is a possibility to postpone by the Ready function.
The minimum number of cycles is 1.

b: It is a memory access cycle, and there is a possibility to postpone by the Ready function. When the immediately succeeding instruction references the register to be subject to LD operation, however, an interlock is applied and the number of execution cycles is incremented by 1.
When the number of uncompleted LD instructions reaches 4, the interlock is applied, continuing from that point until the first LD instruction is completed, and the number of execution cycles is incremented by a given number (number-of-memory-access cycles-number-of-cycles-from-instruction-issuance-until-completion -of-first-LD instruction).

c: If the succeeding instruction references MDH, an interlock is applied and the number of execution cycles is incremented to become 2. Otherwise, the number of cycles is 1.

d: If no read-ahead instruction has been executed on the prefetch buffer, the number of cycles is 2. The minimum number of cycles is 1.

(6) The flag change is shown.



(7) O is applied for RMW type of instructions.

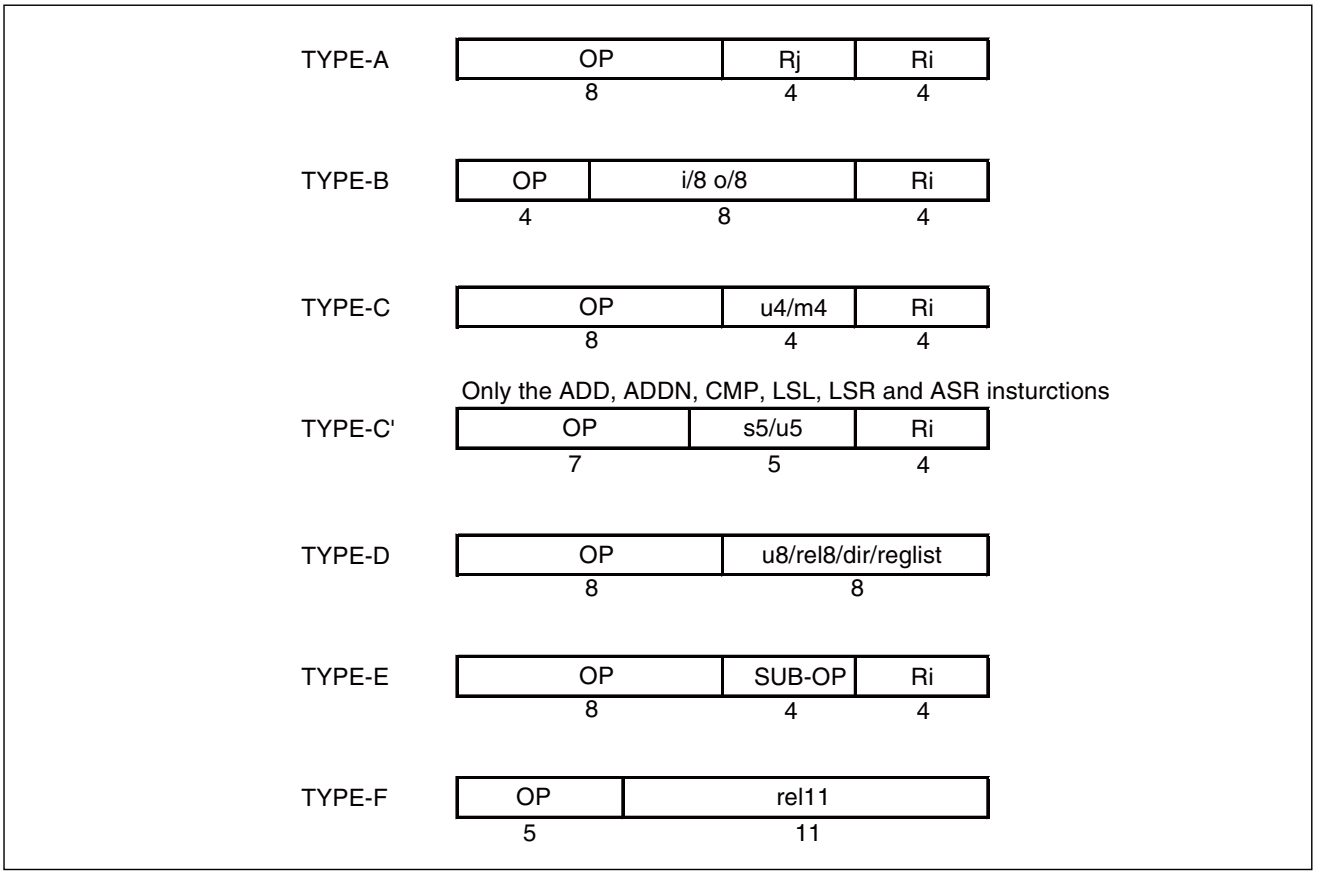
(8) The instruction operation is written.

■ Addressing Mode Symbols

Ri	: register direct (R0 to R15, AC, FP, SP)
Rj	: register direct (R0 to R15, AC, FP, SP)
R13	: register direct (R13, AC)
Ps	: Register direct (program status register)
Rs	: register direct (TBR, RP, SSP, USP, MDH, MDL)
#i4	: 4-bit value immediately (zero externsion:0 to 15, negative extension:-16 to -1)
#i8	: Unsigned 8-bit value immediately (0 to 255)
#i20	: Unsigned 20-bit value immediately (-0X80000 to 0XFFFFF) Attention: -0X7FFFF to -1 is treated as 0X7FFFF to 0XFFFFF.
#i32	: Unsigned 32-bit value immediately (-0X80000000 to 0XFFFFFFFF) Attention: 0X80000000 to -1 is treated as 0X80000000 to 0XFFFFFFFF.
#s5	: Signed 5-bit immediate value (-16 to 15)
#s10	: Signed 10-bit immediate value (only multiples of 4, - 512 to 508)
#u4	: Unsigned 4-bit value immediately (0 to 15)
#u5	: Unsigned 5-bit value immediately (0 to 31)
#u8	: Unsigned 8-bit value immediately (0 to 255)
#u10	: Unsigned 10-bit value immediately (Only the multiple of 4, 0 to 1020)
@dir8	: Unsigned 8-bit direct address (0 to 0XFF)
@dir9	: Unsigned 9-bit direct address (Only the multiple of 2, 0 to 0X1FE)
@dir10	: Unsigned 10-bit direct address (Only the multiple of 4, 0 to 0X3FC)
label9	: Signed 9-bit branch address (only multiples of 2, -0X100 to 0XFC)
label12	: Signed 12-bit branch address (only multiples of 2, -0X800 to 0X7FC)
label20	: Signed 20-bit branch address (-0X80000 to 0X7FFFF)

label32	: Signed 32-bit branch address (-0X80000000 to 0X7FFFFFFF)
@Ri	: Register indirect (R0 to R15, AC, FP, SP)
@Rj	: Register indirect (R0 to R15, AC, FP, SP)
@(R13,Rj)	: Relativity is register indirect (Rj: R0 to R15, AC, FP, SP)
(R14,disp10)	: Relative indirectly register (Only the multiple of disp10:-0X200 to 0X1FC 4)
@(R14,disp9)	: Relative indirectly register (Only the multiple of disp9: -0X100 to 0XFE 2)
(R14,disp8)	: Relativity is register indirect (disp8: -0X80 to 0X7F)
@(R15,udisp6)	: Relative indirectly register (Only the multiple of 4, udisp6: 0 to 60)
@Ri+	: Register indirect with post increment (R0 to R15, AC, FP, SP)
@R13+	: Register indirect with post increment (R13, AC)
@SP+	: Stack pop
@-SP	: Stack push
(reglist)	: Register list

■ Instruction Format



■ Operation Column

The symbols listed below are used in the operation column of the instruction tables and in operations for the instruction rules.

extu( )	It represents a zero extension operation. The empty higher bits are padded with "0"s.
extn( )	It represents a negative extension operation. The empty higher bits are padded with "1"s.
exts( )	It represents a signed extension operation. If the MSB of the data in ( ) is "0", a zero extension operation is performed; if the MSB is "1", a negative extension operation is performed.
&	It represents a logical multiplication (AND) of each bit.
|	It represents a logical addition (OR) of each bit.
^	It represents an exclusive disjunction (EXOR) of each bit.
( )	Parentheses indicate indirect addressing. A value is read from or written to memory at the address indicated by the register or expression in ( ).
{ }	Curly brackets explicitly indicate the priority of operations. { } is used because ( ) is used for indirect addressing.

if (condition) then {expression}
or
if (condition) then {expression 1} else {expression 2}
	Each represents conditional execution. If the condition is satisfied, the expression following "then" is executed. If the condition is not satisfied, the expression following "else" is executed. One or more expressions enclosed in { } can be scripted.

[m:n]	Bits are retrieved from bit m to bit n for an operation.

E.2 Instruction Tables

This section explains the instructions for the FR80 family CPUs.

There is a total of 162 instructions for the FR80 family CPUs. They are categorized into the following 15 types:

- Add-subtract instructions
- Comparison operation instructions
- Logical operation instructions
- Bit operation instructions
- Multiplication and division instructions
- Shift operation instructions
- Immediate value data transfer instructions
- Memory load instructions
- Memory store instructions
- Register-to-register transfer instructions/Dedicated register transfer instructions
- Non-delayed branch instructions
- Delayed branch instructions
- Direct addressing instructions
- Bit search instructions
- Other instructions

Table E-1 Add-subtract instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ADD Rj, Ri	A	A6	1	CCCC	-	Ri + Rj → Ri	
*ADD #s5, Ri	C'	-	1	CCCC	-	Ri + s5 → Ri	The upper 1 bit of s5 is considered as a sign in the assembler.
ADD #i4, Ri	C	A4	1	CCCC	-	Ri + extu(i4) → Ri	i4 is a zero extension.
ADD2 #i4, Ri	C	A5	1	CCCC	-	Ri + extn(i4) → Ri	i4 is a negative extension.
ADDC Rj, Ri	A	A7	1	CCCC	-	Ri + Rj + C → Ri	Addition with a carry
ADDN Rj, Ri	A	A2	1	----	-	Ri + Rj → Ri	
*ADDN #s5, Ri	C'	-	1	----	-	Ri + s5 → Ri	The upper 1 bit of s5 is considered as a sign in the assembler.
ADDN #i4, Ri	C	A0	1	----	-	Ri + extu(i4) → Ri	i4 is a zero extension.
ADDN2 #i4, Ri	C	A1	1	----	-	Ri + extn(i4) → Ri	i4 is a negative extension.
SUB Rj, Ri	A	AC	1	CCCC	-	Ri-Rj → Ri	
SUBC Rj, Ri	A	AD	1	CCCC	-	Ri-Rj-C → Ri	Subtraction with a carry
SUBN Rj, Ri	A	AE	1	----	-	Ri-Rj → Ri	

Table E-2 Comparison operation instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
CMP Rj, Ri	A	AA	1	CCCC	-	Ri-Rj	
*CMP #s5, Ri	C'	-	1	CCCC	-	Ri-s5	The upper 1 bit of s5 is considered as a sign in the assembler.
CMP #i4, Ri	C	A8	1	CCCC	-	Ri-extu(i4)	i4 is a zero extension.
CMP2 #i4, Ri	C	A9	1	CCCC	-	Ri-extn(i4)	i4 is a negative extension.

Table E-3 Logical operation instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
AND Rj, Ri	A	82	1	CC--	-	Ri & Rj → Ri	Word
AND Rj, @Ri	A	84	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Word
ANDH Rj, @Ri	A	85	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Half word
ANDB Rj, @Ri	A	86	1 + 2a	CC--	O	(Ri) & Rj → (Ri)	Byte
OR Rj, Ri	A	92	1	CC--	-	Ri | Rj → Ri	Word
OR Rj, @Ri	A	94	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Word
ORH Rj, @Ri	A	95	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Half word
ORB Rj, @Ri	A	96	1 + 2a	CC--	O	(Ri) | Rj → (Ri)	Byte
EOR Rj, Ri	A	9A	1	CC--	-	Ri ^ Rj → Ri	Word
EOR Rj, @Ri	A	9C	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Word
EORH Rj, @Ri	A	9D	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Half word
EORB Rj, @Ri	A	9E	1 + 2a	CC--	O	(Ri) ^ Rj → (Ri)	Byte

Table E-4 Bit operation instructions (1 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
BANDL #u4, @Ri	C	80	1 + 2a	----	O	(Ri) & $\{F0_H + u4\}$ → (Ri)	Lower 4 bits
BANDH #u4, @Ri	C	81	1 + 2a	----	O	(Ri) & $\{u4<<4 + 0F_H\}$ → (Ri)	Higher 4 bits
*BAND #u8, @Ri *1	-	-	-	----	O	(Ri) &= u8	
BORL #u4, @Ri	C	90	1 + 2a	----	O	(Ri) | u4 → (Ri)	Lower 4 bits
BORH #u4, @Ri	C	91	1 + 2a	----	O	(Ri) | {u4<<4} → (Ri)	Higher 4 bits
*BOR #u8, @Ri *2	-	-	-	----	O	(Ri) | = u8	
BEORL #u4, @Ri	C	98	1 + 2a	----	O	(Ri) ^ u4 → (Ri)	Lower 4 bits
BEORH #u4, @Ri	C	99	1 + 2a	----	O	(Ri) ^ {u4<<4} → (Ri)	Higher 4 bits

Table E-4 Bit operation instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
*BEOR #u8, @Ri *3	-	-	-	----	O	(Ri) ^ = u8	
BTSTL #u4, @Ri	C	88	2 + a	0C--	-	(Ri) & u4	Lower 4 bits
BTSTH #u4, @Ri	C	89	2 + a	CC--	-	(Ri) & {u4<<4}	Higher 4 bits

*1 The assembler generates BANDL or BANDH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BANDL and BANDH are occasionally generated.

*2 The assembler generates BORL or BORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BORL and BORH are occasionally generated.

*3 The assembler generates BEORL or BEORH when the bit is set at u8&0x0F or u8&0xF0, respectively. Both BEORL and BEORH are occasionally generated.

Table E-5 Multiplication and division instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MUL Rj, Ri	A	AF	5	CCC-	-	Ri x Rj → MDH,MDL	32 x 32 bits = 64 bits
MULU Rj, Ri	A	AB	5	CCC-	-	Ri x Rj → MDH,MDL	Unsigned
MULH Rj, Ri	A	BF	3	CC--	-	Ri x Rj → MDL	16 x 16 bits = 32 bits
MULUH Rj, Ri	A	BB	3	CC--	-	Ri x Rj → MDL	Unsigned
DIV0S Ri	E	97-4	1	----	-	With the given instruction sequence MDL / Ri → MDL MDL % Ri → MDH	Step operation 32 / 32 bits = 32 bits
DIV0U Ri	E	97-5	1	----	-		
DIV1 Ri	E	97-6	1	-C-C	-		
DIV2 Ri	E	97-7	c	-C-C	-		
DIV3	E	9F-6	1	----	-		
DIV4S	E	9F-7	1	----	-		
*DIV Ri *1	-	-	36	-C-C	-	MDL/Ri → MDL, MDL %Ri → MDH	
*DIVU Ri *2	-	-	36	-C-C	-	MDL/Ri → MDL, MDL %Ri → MDH	

*1 DIV0S, DIV1×32, DIV2, DIV3, and DIV4S are generated. The instruction code length becomes 72 bytes.

*2 DIV0U, DIV1×32 are generated. The instruction code length becomes 66 bytes.

Table E-6 Shift operation instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LSL Rj, Ri	A	B6	1	CC-C	-	Ri << Rj → Ri	Logical shift
*LSL #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri << u5 → Ri	
LSL #u4, Ri	C	B4	1	CC-C	-	Ri << u4 → Ri	
LSL2 #u4, Ri	C	B5	1	CC-C	-	Ri << {u4 + 16} → Ri	
LSR Rj, Ri	A	B2	1	CC-C	-	Ri >> Rj → Ri	Logical shift
*LSR #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri >> u5 → Ri	
LSR #u4, Ri	C	B0	1	CC-C	-	Ri >> u4 → Ri	
LSR2 #u4, Ri	C	B1	1	CC-C	-	Ri >> {u4 + 16} → Ri	
ASR Rj, Ri	A	BA	1	CC-C	-	Ri >> Rj → Ri	Arithmetic shift
*ASR #u5, Ri (u5: 0 to 31)	C'	-	1	CC-C	-	Ri >> u5 → Ri	
ASR #u4, Ri	C	B8	1	CC-C	-	Ri >> u4 → Ri	
ASR2 #u4, Ri	C	B9	1	CC-C	-	Ri >> {u4 + 16} → Ri	

Table E-7 Immediate value data transfer instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LDI:32 #i32, Ri	H	9F-8	d	----	-	i32 → Ri	
LDI:20 #i20, Ri	G	9B	d	----	-	extu(i20) → Ri	The higher 12 bits are a zero extension.
LDI:8 #i8, Ri	B	C0	1	----	-	extu(i8) → Ri	The higher 24 bits are a zero extension.
*LDI {i8 | i20 | i32}, Ri *1	-	-	-	-	-	{i8 | i20 | i32} → Ri	

*1 When the immediate value is an absolute value, i8, i20, i32 is selected automatically by the assembler.
When the immediate value is a relative value or includes an externally referenced symbol, i32 is selected.

Table E-8 Memory load instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
LD @Rj, Ri	A	04	b	----	-	(Rj) → Ri	Word
LD @(R13, Rj), Ri	A	00	b	----	-	(R13 + Rj) → Ri	
LD @(R14, disp10), Ri	B	2	b	----	-	(R14 + o8 x 4) → Ri	
LD @(R15, udisp6), Ri	C	03	b	----	-	(R15 + u4 x 4) → Ri	
LD @R15+, Ri	E	07-0	b	----	-	(R15) → Ri, R15 + 4 → R15	
LD @R15+, Rs	E	07-8	b	----	-	(R15) → Rs, R15 + 4 → R15	Rs : Special register
LD @R15+, PS	E	07-9	1 + a	CCCC	-	(R15) → PS, R15 + 4 → R15	Word
LDUH @Rj, Ri	A	05	b	----	-	extu((Rj)) → Ri	Half word zero extension
LDUH @(R13, Rj), Ri	A	01	b	----	-	extu((R13 + Rj)) → Ri	
LDUH @(R14, disp9), Ri	B	04	b	----	-	extu((R14 + o8 x 2)) → Rj	
LDUB @Rj, Ri	A	06	b	----	-	extu((Rj)) → Ri	Byte zero extension
LDUB @(R13, Rj), Ri	A	02	b	----	-	extu((R13 + Rj)) → Ri	
LDUB @(R14, disp8), Ri	B	6	b	----	-	extu((R14 + o8)) → Ri	

- The relationship between the instruction format TYPE-B o8/TYPE-C u4 fields and disp8 to disp10 in assembler code is as follows:

  o8 = disp8

  o8 = disp9 >> 1

  o8 = disp10 >> 2

  u4 = udisp6 >> 2

Table E-9 Memory store instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ST Ri, @Rj	A	14	a	----	-	Ri → (Rj)	Word
ST Ri, @(R13, Rj)	A	10	a	----	-	Ri → (R13 + Rj)	
ST Ri, @(R14, disp10)	B	3	a	----	-	Ri → (R14 + o8 x 4)	
ST Ri, @(R15, udisp6)	C	13	a	----	-	Ri → (R15 + u4 x 4)	
ST Ri, @-R15	E	17-0	a	----	-	R15 - 4 → R15, Ri → (R15)	
ST Rs, @-R15	E	17-8	a	----	-	R15 - 4 → R15, Rs → (R15)	Rs : Special register
ST PS, @-R15	E	17-9	a	----	-	R15 - 4 → R15, PS → (R15)	Word
STH Ri, @Rj	A	15	a	----	-	Ri → (Rj)	Half word
STH Ri, @(R13, Rj)	A	11	a	----	-	Ri → (R13 + Rj)	
STH Ri, @(R14, disp9)	B	5	a	----	-	Ri → (R14 + o8 x 2)	
STB Ri, @Rj	A	16	a	----	-	Ri → (Rj)	Byte
STB Ri, @(R13, Rj)	A	12	a	----	-	Ri → (R13 + Rj)	
STB Ri, @(R14, disp8)	B	7	a	----	-	Ri → (R14 + o8)	

- The relationship between the instruction format TYPE-B o8/TYPE-C u4 fields and disp8 to disp10 in assembler code is as follows:

  o8 = disp8

  o8 = disp9 >> 1

  o8 = disp10 >> 2

  u4 = udisp6 >> 2

Table E-10 Register-to-register transfer instructions/dedicated register transfer instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
MOV Rj, Ri	A	8B	1	----	-	Rj → Ri	Transfer between general-purpose registers
MOV Rs, Ri	A	B7	1	----	-	Rs → Ri	Rs: special register
MOV Ri, Rs	A	B3	1	----	-	Ri → Rs	Rs: special register
MOV PS, Ri	E	17-1	1	----	-	PS → Ri	PS: Program status
MOV Ri, PS	E	07-1	1	CCCC	-	Ri → PS	PS: Program status

Table E-11 Non-delayed branch instructions (1 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
JMP @Ri	E	97-0	2	----	-	Ri → PC
CALL label12	F	D0	2	----	-	PC + 2 → RP, PC + 2 + exts(rel11 x 2) → PC
CALL @Ri	E	97-1	2	----	-	PC + 2 → RP, Ri → PC
RET	E	97-2	2	----	-	RP → PC
INT #u8	D	1F	1 + 3a	----	-	SSP-4 → SSP, PS → (SSP), SSP-4 → SSP, PC + 2 → (SSP), 0 → CCR:I, 0 → CCR:S, (TBR + 3FC-u8 x 4) → PC
INTE	E	9F-3	1 + 3a	----	-	SSP-4 → SSP, PS → (SSP), SSP-4 → SSP, PC + 2 → (SSP), 0 → CCR:S, 4 → ILM, (TBR + 3D8) → PC
RETI	E	97-3	1 + 2b	----	-	(SSP) → PC, SSP + 4 → SSP, (SSP) → PS, SSP + 4 → SSP
BRA label9	D	E0	2	----	-	PC + 2 + exts(rel8 x 2) → PC
BNO label9	D	E1	1	----	-	Non-branch
BEQ label9	D	E2	2/1	----	-	if (Z==1) then PC + 2 + exts(rel8 x 2) → PC
BNE label9	D	E3	2/1	----	-	if (Z==0) then PC + 2 + exts(rel8 x 2) → PC
BC label9	D	E4	2/1	----	-	if (C==1) then PC + 2 + exts(rel8 x 2) → PC
BNC label9	D	E5	2/1	----	-	if (C==0) then PC + 2 + exts(rel8 x 2) → PC
BN label9	D	E6	2/1	----	-	if (N==1) then PC + 2 + exts(rel8 x 2) → PC
BP label9	D	E7	2/1	----	-	if (N==0) then PC + 2 + exts(rel8 x 2) → PC
BV label9	D	E8	2/1	----	-	if (V==1) then PC + 2 + exts(rel8 x 2) → PC
BNV label9	D	E9	2/1	----	-	if (V==0) then PC + 2 + exts(rel8 x 2) → PC
BLT label9	D	EA	2/1	----	-	if (V ^ N==1) then PC + 2 + exts(rel8 x 2) → PC
BGE label9	D	EB	2/1	----	-	if (V ^ N==0) then PC + 2 + exts(rel8 x 2) → PC
BLE label9	D	EC	2/1	----	-	if ({V ^ N} | Z==1) then PC + 2 + exts(rel8 x 2) → PC
BGT label9	D	ED	2/1	----	-	if ({V ^ N} | Z==0) then PC + 2 + exts(rel8 x 2) → PC

Table E-11 Non-delayed branch instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
BLS label9	D	EE	2/1	----	-	if (C or Z==1) then PC + 2 + exts(rel8 x 2) → PC
BHI label9	D	EF	2/1	----	-	if (C or Z==0) then PC + 2 + exts(rel8 x 2) → PC

- "2/1" in the CYC column represents 2 in cases of branching and 1 in cases of no branching.
- The stack flag (S) must be "0" when RETI is executed.
- The relationship between the instruction format TYPE-D rel8/TYPE-F rel11 fields and label9/label12 in assembler code is as follows:

  rel8 = (label9-PC-2)/2

  rel11 = (label12-PC-2)/2

Table E-12 Delayed branch instructions (1 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
JMP:D @Ri	E	9F-0	1	----	-	Ri → PC
CALL:D label12	F	D8	1	----	-	PC + 4 → RP, PC + 2 + exts(rel11 x 2) → PC
CALL:D @Ri	E	9F - 1	1	----	-	PC + 4 → RP, Ri → PC
RET:D	E	9F - 2	1	----	-	RP → PC
BRA:D label9	D	F0	1	----	-	PC + 2 + exts(rel8 x 2) → PC
BNO:D label9	D	F1	1	----	-	Non-branch
BEQ:D label9	D	F2	1	----	-	if (Z==1) then PC + 2 + exts(rel8 x 2) → PC
BNE:D label9	D	F3	1	----	-	if (Z==0) then PC + 2 + exts(rel8 x 2) → PC
BC:D label9	D	F4	1	----	-	if (C==1) then PC + 2 + exts(rel8 x 2) → PC
BNC:D label9	D	F5	1	----	-	if (C==0) then PC + 2 + exts(rel8 x 2) → PC
BN:D label9	D	F6	1	----	-	if (N==1) then PC + 2 + exts(rel8 x 2) → PC
BP:D label9	D	F7	1	----	-	if (N==0) then PC + 2 + exts(rel8 x 2) → PC
BV:D label9	D	F8	1	----	-	if (V==1) then PC + 2 + exts(rel8 x 2) → PC
BNV:D label9	D	F9	1	----	-	if (V==0) then PC + 2 + exts(rel8 x 2) → PC
BLT:D label9	D	FA	1	----	-	if (V ^ N==1) then PC + 2 + exts(rel8 x 2) → PC

Table E-12 Delayed branch instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation
BGE:D label9	D	FB	1	----	-	if (V ^ N==0) then PC + 2 + exts(rel8 x 2) → PC
BLE:D label9	D	FC	1	----	-	if ({V ^ N} | Z==1) then PC + 2 + exts(rel8 x 2) → PC
BGT:D label9	D	FD	1	----	-	if ({V ^ N} | Z==0) then PC + 2 + exts(rel8 x 2) → PC
BLS:D label9	D	FE	1	----	-	if (C or Z==1) then PC + 2 + exts(rel8 x 2) → PC
BHI:D label9	D	FF	1	----	-	if (C or Z==0) then PC + 2 + exts(rel8 x 2) → PC

- Branching for a delayed branch instruction occurs after the next instruction (delay slot) is executed.
- The relationship between the instruction format TYPE-D rel8/TYPE-F rel11 fields and label9/label12 in assembler code is as follows:

  rel8 = (label9-PC-2)/2

  rel11 = (label12-PC-2)/2

Table E-13 Direct addressing instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
DMOV @dir10, R13	D	08	b	----	-	(dir10) → R13	Word
DMOV R13, @dir10	D	18	a	----	-	R13 → (dir10)	
DMOV @dir10, @R13+	D	0C	1 + 2a	----	-	(dir10) → (R13), R13+=4	
DMOV @R13+, @dir10	D	1C	1 + 2a	----	-	(R13) → (dir10), R13+=4	
DMOV @dir10, @-R15	D	0B	1 + 2a	----	-	R15-=4, (R15) → (dir10)	
DMOV @R15+, @dir10	D	1B	1 + 2a	----	-	(R15) → (dir10), R15+=4	
DMOVH @dir9, R13	D	09	b	----	-	(dir9) → R13	Half word
DMOVH R13, @dir9	D	19	a	----	-	R13 → (dir9)	
DMOVH @dir9, @R13+	D	0D	1 + 2a	----	-	(dir9) → (R13), R13+=2	
DMOVH @R13+, @dir9	D	1D	1 + 2a	----	-	(R13) → (dir9), R13+=2	
DMOVB @dir8, R13	D	0A	b	----	-	(dir8) → R13	Byte
DMOVB R13, @dir8	D	1A	a	----	-	R13 → (dir8)	
DMOVB @dir8, @R13+	D	0E	1 + 2a	----	-	(dir8) → (R13), R13++	
DMOVB @R13+, @dir8	D	1E	1 + 2a	----	-	(R13) → (dir8), R13++	

- The relationship between the instruction format TYPE-D dir8 field and dir8, dir9, and dir10 in assembler code is as follows:

  dir8 = dir8

  dir8 = dir9 >> 1

  dir8 = dir10 >> 2

Table E-14 Bit search instructions

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
SRCH0 Ri	E	97-C	1	----	-	search_zero (Ri) → Ri	Searches for the first 0 bit from MSB to LSB
SRCH1 Ri	E	97-D	1	----	-	search_one (Ri) → Ri	Searches for the first 1 bit from MSB to LSB
SRCHC Ri	E	97-E	1	----	-	search_change (Ri) → Ri	Searches for the first change from MSB to LSB

Table E-15 Other instructions (1 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
NOP	E'	9F-A	1	----	-	No change	
ANDCCR #u8	D	83	1	CCCC	-	CCR & u8 → CCR	
ORCCR #u8	D	93	1	CCCC	-	CCR | u8 → CCR	
STILM #u8	D	87	1	----	-	u8 → ILM	ILM immediate value set
ADDSP #s10	D	A3	1	----	-	R15 += s10	
EXTSB Ri	E	97-8	1	----	-	exts (Ri[7:0]) → Ri	Signed extension 8 → 32
EXTUB Ri	E	97-9	1	----	-	extu (Ri[7:0]) → Ri	Zero extension 8 → 32
EXTSH Ri	E	97-A	1	----	-	exts (Ri[15:0]) → Ri	Signed extension 16 → 32
EXTUH Ri	E	97-B	1	----	-	extu (Ri[15:0]) → Ri	Zero extension 16 → 32
LDM0(reglist)	D	8C	*1	----	-	(R15) → reglist, R15 increment	Load Multi R0 to R7
LDM1(reglist)	D	8D	*1	----	-	(R15) → reglist, R15 increment	Load Multi R8 to R15
*LDM(reglist)[*3]	-	-	-	----	-	(R15) → reglist, R15 increment	Load Multi R0 to R15
STM0(reglist)	D	8E	*2	----	-	R15 decrement, reglist → (R15)	Store Multi R0 to R7
STM1(reglist)	D	8F	*2	----	-	R15 decrement, reglist → (R15)	Store Multi R8 to R15
*STM(reglist)[*4]	-		-	----	-	R15 decrement, reglist → (R15)	Store Multi R0 to R15

Table E-15 Other instructions (2 / 2)

Mnemonic	Type	OP	CYC	FLAG NZVC	RMW	Operation	Remarks
ENTER #u10	D	0F	1 + a	----	-	R14 → (R15-4), R15 - 4 → R14, R15-extu (u8 x 4) → R15	Function entry processing
LEAVE	E	9F - 9	b	----	-	R14 + 4 → R15, (R15-4) → R14	Function exit processing
XCHB @Rj, Ri	A	8A	2a	----	O	Ri → TEMP, extu((Rj)) → Ri, TEMP → (Rj)	Byte data for semaphore management

*1 : The number of execution cycles for LDM0 (reglist) and LDM1 (reglist) becomes b x n cycles when the number of registers specified is n.

*2 : The number of execution cycles for STM0 (reglist) and STM1 (reglist) becomes a x n cycles when the number of registers specified is n.

*3 : If reglist specifies any of R0 to R7, LDM0 is generated.
If it specifies any of R8 to R15, LDM1 is generated. Both LDM0 and LDM1 are occasionally generated.

*4 : If reglist specifies any of R0 to R7, STM0 is generated.
If it specifies any of R8 to R15, STM1 is generated. Both STM1 and STM0 are occasionally generated.

- In the ADDSP instruction, the relationship between the instruction format TYPE-D s8 field and s10 in assembler code is as follows:

  s8 = s10 >> 2

- In the ENTER instruction, the relationship between the instruction format TYPE-D u8 field and u10 in assembler code is as follows:

  u8 = u10 >> 2

E.3 List of Instructions That Can Be Specified for Delay Slots

This section lists instructions that can be specified for delay slots in delayed branch instructions.

- Add-subtract instructions

ADD Rj, Ri	ADD #14, Ri	ADD2 #i4, Ri
ADDC Rj, Ri	ADDN Rj, Ri	ADDN #i4, Ri
ADDN2 #i4, Ri	SUB Rj, Ri	SUBC Rj, Ri
SUBN Rj, Ri		

- Comparison operation instructions

CMP Rj, Ri	CMP #i4, Ri	CMP2 #i4, Ri

- Logical operation instructions

AND Rj, Ri	OR Rj, Ri	EOR Rj, Ri

- Multiplication and division instructions

DIV0S Ri	DIV0U Ri	DIV1 Ri
DIV2 Ri	DIV3	DIV4S

- Shift operation instructions

LSL Rj, Ri	LSL #u4, Ri	LSL2 #u4, Ri
LSR Rj, Ri	LSR #u4, Ri	LSR2 #u4, Ri
ASR Rj, Ri	ASR #u4, Ri	ASR2 #u4, Ri

- Immediate value data transfer instruction

LDI:8 #i8, Ri

- Memory load instructions

LD @Rj, Ri	LD @(R13, Rj), Ri	LD @(R14, disp10), Ri
LD @(R15, udisp6), Ri	LD @R15+, Ri	LD @R15+, Rs
LDUH @Rj, Ri	LDUH @(R13, Rj), Ri	LDUH @(R14, disp9), Ri
LDUB @Rj, Ri	LDUB @(R13, Rj), Ri	LDUB @(R14, disp8), Ri

- Memory store instructions

ST Ri, @Rj	ST Ri, @(R13, Rj)	ST Ri, @(R14, disp10)
ST Ri, @(R15, udisp6)	ST Ri, @-R15	ST Rs, @-R15
ST PS, @-R15		
STH Ri, @Rj	STH Ri, @(R13, Rj)	STH Ri, @(R14, disp9)
STB Ri, @Rj	STB Ri, @(R13, Rj)	STB Ri, @(R14, disp8)

- Register-to-register transfer instructions

MOV Rj, Ri	MOV Rs, Ri	MOV Ri, Rs
MOV PS, Ri	MOV Ri, PS	

- Direct addressing instructions

DMOV @dir10, R13	DMOV R13, @dir10	DMOVH @dir9, R13
DMOVH R13, @dir9	DMOVB @dir8, R13	DMOVB R13, @dir8

- Bit search instructions

SRCH0 Ri	SRCH1 Ri	SRCHC Ri

- Other instructions

NOP	ANDCCR #u8	ORCCR #u8
STILM #u8	ADDSP #s10	EXTSB Ri
EXTUB Ri	EXTSH Ri	EXTUH Ri
LEAVE		

Index of Pins

Pin Name	Related Function	Page
A		
A00	External bus interface	300
A01	External bus interface	300
A02	External bus interface	300
A03	External bus interface	300
A04	External bus interface	300
A05	External bus interface	300
A06	External bus interface	300
A07	External bus interface	300
A08	External bus interface	300
A09	External bus interface	300
A10	External bus interface	300
A11	External bus interface	300
A12	External bus interface	300
A13	External bus interface	300
A14	External bus interface	300
A15	External bus interface	300
A16	External bus interface	300
A17	External bus interface	300
A18	External bus interface	300
A19	External bus interface	300
A20	External bus interface	300
A21	External bus interface	300
A22	External bus interface	300
A23	External bus interface	300
ADTRG0	10-bit A/D converter	777
AIN0	Up/down counter	744
AIN1	Up/down counter	744
AIN2	Up/down counter	744
AIN3	Up/down counter	744
AN0	10-bit A/D converter	777
AN1	10-bit A/D converter	777
AN2	10-bit A/D converter	777
AN3	10-bit A/D converter	777
AN4	10-bit A/D converter	777
AN5	10-bit A/D converter	777
AN6	10-bit A/D converter	777
AN7	10-bit A/D converter	777
AN8	10-bit A/D converter	777
AN9	10-bit A/D converter	777
AN10	10-bit A/D converter	777
AN11	10-bit A/D converter	777

Pin Name	Related Function	Page
AN12	10-bit A/D converter	777
AN13	10-bit A/D converter	777
AN14	10-bit A/D converter	777
AN15	10-bit A/D converter	777
AN16	10-bit A/D converter	777
AN17	10-bit A/D converter	777
AN18	10-bit A/D converter	777
AN19	10-bit A/D converter	777
AN20	10-bit A/D converter	777
AN21	10-bit A/D converter	777
AN22	10-bit A/D converter	777
AN23	10-bit A/D converter	777
$\overline{AS}$	External bus interface	300
AV_{CC}	10-bit A/D converter	777
AVRH	10-bit A/D converter	777
AV_{SS}	10-bit A/D converter	777
B		
BIN0	Up/down counter	744
BIN1	Up/down counter	744
BIN2	Up/down counter	744
BIN3	Up/down counter	744
C		
$\overline{CS0}$	External bus interface	300
$\overline{CS1}$	External bus interface	300
$\overline{CS2}$	External bus interface	300
$\overline{CS3}$	External bus interface	300
D		
D00	External bus interface	300
D01	External bus interface	300
D02	External bus interface	300
D03	External bus interface	300
D04	External bus interface	300
D05	External bus interface	300
D06	External bus interface	300
D07	External bus interface	300
D08	External bus interface	300
D09	External bus interface	300
D10	External bus interface	300
D11	External bus interface	300
D12	External bus interface	300
D13	External bus interface	300
D14	External bus interface	300
D15	External bus interface	300
DA0	8-bit D/A converter	853
DA1	8-bit D/A converter	853
DA2	8-bit D/A converter	853

Pin Name	Related Function	Page
DACK0	DMA controller (DMAC) / External bus interface	1187, 300
DACK1	DMA controller (DMAC) / External bus interface	1187, 300
DACK2	DMA controller (DMAC) / External bus interface	1187, 300
DACK3	DMA controller (DMAC) / External bus interface	1187, 300
DEOP0	DMA controller (DMAC) / External bus interface	1187, 300
DEOP1	DMA controller (DMAC) / External bus interface	1187, 300
DEOP2	DMA controller (DMAC) / External bus interface	1187, 300
DEOP3	DMA controller (DMAC) / External bus interface	1187, 300
DREQ0	DMA controller (DMAC) / External bus interface	1187, 300
DREQ1	DMA controller (DMAC) / External bus interface	1187, 300
DREQ2	DMA controller (DMAC) / External bus interface	1187, 300
DREQ3	DMA controller (DMAC) / External bus interface	1187, 300
F		
FRCK0	32-bit free-run timer	523
FRCK1	32-bit free-run timer	523
I		
IN0	32-bit input capture	541
IN1	32-bit input capture	541
IN2	32-bit input capture	541
IN3	32-bit input capture	541
IN4	32-bit input capture	541
IN5	32-bit input capture	541
IN6	32-bit input capture	541
IN7	32-bit input capture	541
$\overline{\text{INIT}}$	Reset	231
INT0	External interrupt request	475
INT1	External interrupt request	475
INT2	External interrupt request	475
INT3	External interrupt request	475
INT4	External interrupt request	475
INT5	External interrupt request	475
INT6	External interrupt request	475
INT7	External interrupt request	475
INT8	External interrupt request	475
INT9	External interrupt request	475
INT10	External interrupt request	475
INT11	External interrupt request	475
INT12	External interrupt request	475
INT13	External interrupt request	475
INT14	External interrupt request	475
INT15	External interrupt request	475
INT16	External interrupt request	475
INT17	External interrupt request	475
INT18	External interrupt request	475
INT19	External interrupt request	475
INT20	External interrupt request	475
INT21	External interrupt request	475

Pin Name	Related Function	Page
INT22	External interrupt request	475
INT23	External interrupt request	475
INT24	External interrupt request	475
INT25	External interrupt request	475
INT26	External interrupt request	475
INT27	External interrupt request	475
INT28	External interrupt request	475
INT29	External interrupt request	475
INT30	External interrupt request	475
INT31	External interrupt request	475
M		
MD0	Mode pin	25
MD1	Mode pin	25
O		
OUT0	32-bit output compare	557
OUT1	32-bit output compare	557
OUT2	32-bit output compare	557
OUT3	32-bit output compare	557
OUT4	32-bit output compare	557
OUT5	32-bit output compare	557
OUT6	32-bit output compare	557
OUT7	32-bit output compare	557
P		
P00	I/O port	433
P01	I/O port	433
P02	I/O port	433
P03	I/O port	433
P04	I/O port	433
P05	I/O port	433
P06	I/O port	433
P07	I/O port	433
P10	I/O port	433
P11	I/O port	433
P12	I/O port	433
P13	I/O port	433
P14	I/O port	433
P15	I/O port	433
P16	I/O port	433
P17	I/O port	433
P20	I/O port	433
P21	I/O port	433
P22	I/O port	433
P23	I/O port	433
P24	I/O port	433
P25	I/O port	433
P26	I/O port	433

Pin Name	Related Function	Page
P27	I/O port	433
P30	I/O port	433
P31	I/O port	433
P32	I/O port	433
P33	I/O port	433
P34	I/O port	433
P35	I/O port	433
P36	I/O port	433
P37	I/O port	433
P40	I/O port	433
P41	I/O port	433
P42	I/O port	433
P43	I/O port	433
P44	I/O port	433
P45	I/O port	433
P46	I/O port	433
P47	I/O port	433
P50	I/O port	433
P51	I/O port	433
P52	I/O port	433
P53	I/O port	433
P54	I/O port	433
P55	I/O port	433
P56	I/O port	433
P57	I/O port	433
P60	I/O port	433
P61	I/O port	433
P62	I/O port	433
P63	I/O port	433
P64	I/O port	433
P65	I/O port	433
P66	I/O port	433
P67	I/O port	433
P70	I/O port	433
P71	I/O port	433
P72	I/O port	433
P73	I/O port	433
P74	I/O port	433
P75	I/O port	433
P76	I/O port	433
P77	I/O port	433
P80	I/O port	433
P81	I/O port	433
P82	I/O port	433
P83	I/O port	433
P84	I/O port	433
P85	I/O port	433

Pin Name	Related Function	Page
P86	I/O port	433
P87	I/O port	433
P90	I/O port	433
P91	I/O port	433
P92	I/O port	433
PA0	I/O port	433
PA1	I/O port	433
PA2	I/O port	433
PA3	I/O port	433
PA4	I/O port	433
PA5	I/O port	433
PA6	I/O port	433
PA7	I/O port	433
PG0	I/O port	433
PG1	I/O port	433
PG2	I/O port	433
PG3	I/O port	433
PG4	I/O port	433
PG5	I/O port	433
PG6	I/O port	433
PG7	I/O port	433
PH0	I/O port	433
PH1	I/O port	433
PH2	I/O port	433
PH3	I/O port	433
PK0	I/O port	433
PK1	I/O port	433
PK2	I/O port	433
PK3	I/O port	433
R		
RDY	External bus interface	300
$\overline{RD}$	External bus interface	300
S		
SCK0	Multifunction serial interface	859
SCK1	Multifunction serial interface	859
SCK2	Multifunction serial interface	859
SCK3	Multifunction serial interface	859
SCK4	Multifunction serial interface	859
SCK5	Multifunction serial interface	859
SCK6	Multifunction serial interface	859
SCK7	Multifunction serial interface	859
SCK8	Multifunction serial interface	859
SCK9	Multifunction serial interface	859
SCK10	Multifunction serial interface	859
SCK11	Multifunction serial interface	859
SIFA	Slave Interface	237
$\overline{SIFCS}$	Slave Interface	237

Pin Name	Related Function	Page
SIFD0	Slave Interface	237
SIFD1	Slave Interface	237
SIFD2	Slave Interface	237
SIFD3	Slave Interface	237
SIFD4	Slave Interface	237
SIFD5	Slave Interface	237
SIFD6	Slave Interface	237
SIFD7	Slave Interface	237
SIFD8	Slave Interface	237
SIFD9	Slave Interface	237
SIFD10	Slave Interface	237
SIFD11	Slave Interface	237
SIFD12	Slave Interface	237
SIFD13	Slave Interface	237
SIFD14	Slave Interface	237
SIFD15	Slave Interface	237
$\overline{\text{SIFDRQR}}$	Slave Interface	237
$\overline{\text{SIFDRQT}}$	Slave Interface	238
$\overline{\text{SIFRD}}$	Slave Interface	237
$\overline{\text{SIFWR}}$	Slave Interface	237
SIN0	Multifunction serial interface	859
SIN1	Multifunction serial interface	859
SIN2	Multifunction serial interface	859
SIN3	Multifunction serial interface	859
SIN4	Multifunction serial interface	859
SIN5	Multifunction serial interface	859
SIN6	Multifunction serial interface	859
SIN7	Multifunction serial interface	859
SIN8	Multifunction serial interface	859
SIN9	Multifunction serial interface	859
SIN10	Multifunction serial interface	859
SIN11	Multifunction serial interface	859
SOUT0	Multifunction serial interface	859
SOUT1	Multifunction serial interface	859
SOUT2	Multifunction serial interface	859
SOUT3	Multifunction serial interface	859
SOUT4	Multifunction serial interface	859
SOUT5	Multifunction serial interface	859
SOUT6	Multifunction serial interface	859
SOUT7	Multifunction serial interface	859
SOUT8	Multifunction serial interface	859
SOUT9	Multifunction serial interface	859
SOUT10	Multifunction serial interface	859
SOUT11	Multifunction serial interface	859
SYSCLK	External bus interface	300

Pin Name	Related Function	Page
T		
TIOA0	Base timer	614
TIOA1	Base timer	614
TIOA2	Base timer	614
TIOA3	Base timer	614
TIOA4	Base timer	614
TIOA5	Base timer	614
TIOA6	Base timer	614
TIOA7	Base timer	614
TIOA8	Base timer	614
TIOA9	Base timer	614
TIOA10	Base timer	614
TIOA11	Base timer	614
TIOA12	Base timer	614
TIOA13	Base timer	614
TIOA14	Base timer	614
TIOA15	Base timer	614
TIOB0	Base timer	614
TIOB1	Base timer	614
TIOB2	Base timer	614
TIOB3	Base timer	614
TIOB4	Base timer	614
TIOB5	Base timer	614
TIOB6	Base timer	614
TIOB7	Base timer	614
TIOB8	Base timer	614
TIOB9	Base timer	614
TIOB10	Base timer	614
TIOB11	Base timer	614
TIOB12	Base timer	614
TIOB13	Base timer	614
TIOB14	Base timer	614
TIOB15	Base timer	614
TMI0	16-bit reload timer	577
TMI1	16-bit reload timer	577
TMI2	16-bit reload timer	577
TMO0	16-bit reload timer	577
TMO1	16-bit reload timer	577
TMO2	16-bit reload timer	577
U		
UDM	USB function in Mini-HOST	15
UDP	USB function in Mini-HOST	15
V		
V_{CC}	Power pin	15, 19, 25, 30, 37
V_{SS}	Power pin	13, 15, 19, 24, 30

Pin Name	Related Function	Page
W		
$\overline{WR0}$	External bus interface	300
$\overline{WR1}$	External bus interface	300
X		
X0	Clock control part	139
X0A	Clock control part	139
X1	Clock control part	139
X1A	Clock control part	139
Z		
ZIN0	Up/down counter	744
ZIN1	Up/down counter	744
ZIN2	Up/down counter	744
ZIN3	Up/down counter	744

CM71-10146-1E

FUJITSU MICROELECTRONICS • CONTROLLER MANUAL

FR80

32-BIT MICROCONTROLLER

MB91660 Series

HARDWARE MANUAL

June 2008 the first edition

Published **FUJITSU MICROELECTRONICS LIMITED**

Edited Strategic Business Development Dept.